Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2010 Proceedings

Americas Conference on Information Systems (AMCIS)

8-2010

IT Project Retrospectives: Learning from the Past through a Program of Action Research

R. Ryan Nelson
University of Virginia, rnelson@virginia.edu

IT Project Retrospectives: Learning from the Past through a Program of Action Research

R. Ryan Nelson
McIntire School of Commerce
University of Virginia
rnelson@virginia.edu

ABSTRACT

Project management has emerged as a strong discipline practiced by highly trained, certified professionals as organizations have come to realize they cannot stay in business if they cannot manage their projects effectively. However, most companies are still unable or unwilling to perform the most basic of continuous improvement activities – identifying and learning from past mistakes and successes. To help address this shortcoming, this paper provides a framework for conducting project retrospectives that has evolved through the analysis of 130 IT projects over the past ten years. By integrating the findings of several previous studies, the paper provides guidance on mapping project momentum, evaluating project success, identifying and avoiding classic mistakes through best practices, performing root cause analysis, and delivering actionable recommendations.

Keywords

Project management, retrospectives, postmortems, post-implementation review, success, failure, classic mistakes, best practices, root cause analysis, momentum, action research.

INTRODUCTION

Failure to learn from past mistakes and successes has consistently been a major obstacle to improving IT project management. For example:

> *"We talk about software engineering but reject one of the most basic engineering practices; identifying and learning from our mistakes."* – Boddie (1987)
>
> *"Most of us pay lip service to the need for [IT] project postmortems, but the literature provides little guidance on how to conduct them."* – Collier, DeMarco, and Fearey (1996)
>
> *"The problem is that, like the weather, everyone talks about project retrospectives being a good thing, but no one ever does anything about them."* – Glass (2002)
>
> *"In the information systems (IS) literature, it has been argued that organizations should attempt to learn from their failures. Yet, there is evidence that organizations not only 'fail to learn,' but also 'learn to fail' in the context of IT projects."* – Kasi, Keil, Mathiassen, and Pedersen (2008)

This paper addresses this perpetual shortcoming by integrating, updating, and extending the research findings from three previous studies on IT project retrospectives: Nelson (2005), Nelson (2007), and Nelson and Jansen (2009). The result is a "meta-retrospective" of 130 IT projects analyzed as part of a program of action research conducted between 1999 and 2009. Following a brief discussion on retrospectives, the paper is organized much like a typical retrospective: context and data collection, time lines and momentum maps, evaluation of success/failure, lessons learned, root cause analysis, and recommendations for the future.

IT PROJECT RETROSPECTIVES

A retrospective is a formal method for evaluating project performance, extracting lessons learned, and making recommendations for the future.

Why Retrospectives Are Important

Retrospectives offer a variety of potential benefits, including the following:

- Organizational learning – Getting the collective story out (synergistic learning) and ensuring that individual stakeholders hear the whole story, not just their personal experience.
- Continuous improvement – Facilitating improvements in processes, procedures, and culture.
- Estimation and scheduling – Capture actual data on size, effort, and time for use in calibrating future estimation models and practices.
- Team building – Acknowledge and repair relationship issues as appropriate.
- Recognition and reflection – Pause and reflect on accomplishments before proceeding to "solve the next problem."

Based on these benefits, retrospectives should be expected to at a minimum break even (Kerth, 2001); e.g., a one-day retrospective will save you at least one day on the next project, and many more across multiple projects.

Why Retrospectives Aren't Done

Despite their obvious potential to yield significant benefits, most organizations rarely conduct formal retrospectives, outside of the military (Kasi et al., 2008). According to the Gartner Group, only 13% of their clients conduct such reviews (Hoffman, 2005). The most obvious reason is the natural human desire to put the past to rest and go on to something new (Hayes, 2008).

Enterprises are also reluctant to allocate additional time and money to a project after the system is completed (Bjørnson, Wang, and Arisholm, 2009; Glass, 2002). This reluctance is particularly profound if the project is seen as a failure and often surfaces in a request that the benefits of the retrospective be quantified before it is approved.

Finally, most retrospectives are poorly done, which doesn't help overcome either the social or the financial obstacles (Kasi et al., 2008). As a project manager in a large consulting firm lamented, "Our post-implementation reviews tend to be witch hunts, where the innocent get punished and the guilty get promoted!" In other cases, retrospectives are seen as only "checklist items." Enterprises conduct them, but do not apply the lessons learned. In these cases, it is certainly difficult to see the value.

To address these perceived shortcomings, many large IT services firms have developed proprietary methodologies for conducting retrospectives. The retrospective process has also recently received attention in the Project Management Institute's (PMI) Body of Knowledge and the Software Engineering Institute's (SEI) Capability Maturity Model.

ACTION RESEARCH ON IT PROJECT RETROSPECTIVES

Action research is a reflective process of progressive problem solving led by individuals working with others in teams or as part of a "community of practice" using data-driven collaborative analysis to understand underlying causes enabling future predictions about personal and organizational change (Lewin, 1946; Reason & Bradbury, 2007). This form of research is particularly well suited for assessing completed milestones/projects and making recommendations going forward.

Since the summer of 1999, the University of Virginia has delivered a Master of Science in the Management of Information Technology (MS MIT) degree program in an executive format to working professionals. Over this time, 712 working professionals have participated; each has had an average of over ten years of experience and direct involvement with at least one major IT project. All 712 participants received instruction in how to conduct action research in the form of project retrospectives using the following framework:[1]

- Project Context and Description – including a review of the project charter, organizational map (i.e., chart of all stakeholders and their reporting relationships), and a detailed description of the data collection process (e.g., artifact analysis, stakeholder interviews, etc.)
- Project Timeline and Momentum Map
- Evaluation of Project Success/Failure

[1] The original framework was derived from IT consulting firms during the 1990s and then adapted over the years based on direct experience (see Nelson, 2005) as well as the work of Kerth (2001); Derby and Schwaber (2006); and Kasi et al. (2008).

- Lessons Learned – an evaluation of what went right and what went wrong during the course of the project, including root cause analysis
- Recommendations for the Future

In partial fulfillment of program requirements, the participants have worked in teams and conducted retrospectives of recently completed IT projects. Thus far, 130 retrospectives have been conducted in 95 different organizations. These projects have ranged from relatively small (several hundred thousand dollars) internally built application development projects to very large (over $100 million) mission-critical applications involving multiple external providers.

When viewed individually, each retrospective tells a unique story and provides a rich understanding of the project management practices taken within a specific context during a specific timeframe. However, when viewed as a whole, these 130 projects provide an incredible opportunity to understand project management practices at a more macro level and generate findings that can be generalized across a wide spectrum of applications and organizations.

For example, the analysis of 72 projects completed through 2005 provided a comprehensive view of the major factors in project success. That study illustrated the importance of evaluating project success from multiple dimensions, as well as from different stakeholder perspectives (Nelson, 2005).

The study reported in Nelson (2007) focused on the lessons learned portion of each retrospective (regardless of whether or not the project was ultimately considered a success). This study yielded very interesting findings on what tended to go wrong with the 99 projects studied through 2006.

Most recently, Nelson and Jansen (2009) reported on the results of mapping the momentum of 51 different projects. This study presented both retrospective observations regarding momentum as well as guidelines for helping managers proactively manage momentum during the course of a project.

Together, these three studies provide a longitudinal meta-retrospective with synergistic qualities. A summary of key findings and recommendations for each major retrospective component follows.

TIME LINES AND MOMENTUM MAPS

In addition to contextual project artifacts (e.g., the project charter and organizational map) and metrics (e.g., cost and requirements specifications), time lines are a critical component in telling the story of a project. The first step in building the time line involves getting all relevant stakeholder groups to identify key events that occurred during the life of the project. One common approach is to ask each stakeholder to label index cards with significant events and then post them along a time line divided into meaningful time periods, moving chronologically from left to right on a meeting room wall. After some time for reflection, stakeholders can discuss associations, patterns, anomalies, and/or disagreements. The final product becomes the platform for the remainder of the assessment.

The next step is to overlay a momentum map (a.k.a. "emotional seismograph;" e.g., Kerth, 2001; Derby and Schwaber, 2006) over the project time line (see Figure 1 for an example). From start to finish, all IT projects experience fluctuations in momentum (i.e., shifting energy or force of movement), and momentum maps can be used retrospectively to graphically depict how the pattern of momentum fluctuations led to a successful and timely project completion, or the point at which a failed project was derailed.

Managing Momentum: Lessons for Project Managers

Visual depictions of trends in IT project momentum over time and reasons for fluctuations provide project managers with opportunities to capitalize on rich process data. For example, momentum maps can be used as a communication tool to collectively make sense of events, learn from various stakeholders' perspectives and identify important triggers of momentum shifts. Nelson and Jansen (2009) highlight key insights into managing project momentum gleaned from the synthesis and qualitative analysis of momentum maps constructed for 51 IT projects. These insights are summarized below.

Stakeholders Mostly Have Convergent Perceptions of Momentum

In general, the various stakeholders of a given project were found to share similar perspectives on momentum—particularly when considering the general shape of the map over time, the slope of the curve and significant changes in direction. In other words, when viewed holistically, different stakeholders tend to have a similar view of the impact that significant events have on project momentum. This finding was especially profound at various points in the project life cycle—e.g., at project kickoff, during the later stages of a project and when there are particularly steep slopes representing dramatic changes in project momentum.

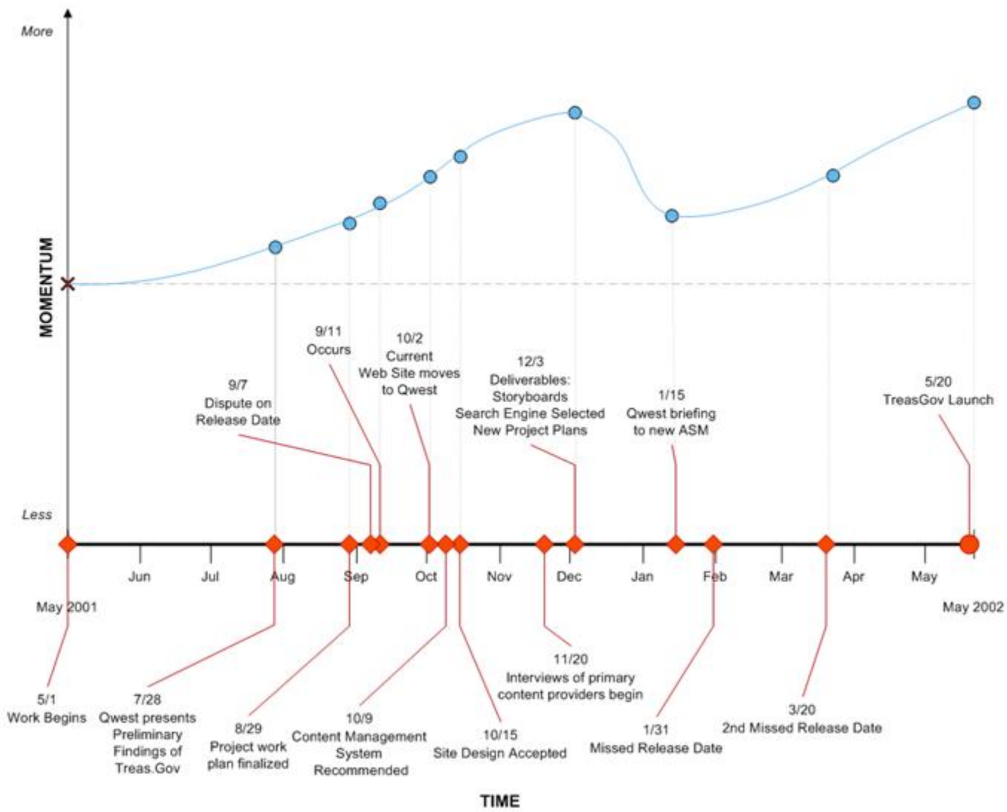

Figure 1. Example of Time Line and Momentum Map

Project Start and Finish

The vast majority of the 51 projects studied (84%) began with an upward trajectory in momentum, indicating that projects tend to start off with significant positive energy. Key events that trigger this positive start are kickoff meetings and announcements, public endorsements by executive sponsors and social interactions. On the other hand, momentum at the end of a project was almost as likely to be on a downward trajectory (45%) as an upward one (55%). Key events that were correlated with declining momentum at the end of a project included missed deadlines and budgets, disenchanted users and "burned out" team members.

Positive and Negative Spirals

Nelson and Jansen also examined the maps for graphical signs of overall increasing or decreasing momentum. About a third (35%) of the projects exhibited a positive (or upward) spiral, defined as having positive trajectories that increasingly outweighed negative dips as the project moved through its lifecycle. A project whose overall momentum remains above the baseline with a positive spiral is perhaps the best case scenario for a project's momentum. In contrast, 12% of projects showed signs of a negative spiral, with negative dips increasingly outweighing positive trajectories over time. Negative spirals represent an overall decrease in energy over time and possibly also indicate that stakeholder commitment declines as the project progresses. The maps of the remaining 53% of projects exhibited no clear overall trend up or down.

Inflection Points

The most interesting features of the momentum maps are the inflection points—where there is a significant change in the direction of project momentum. On average, projects experienced a total of five significant changes in momentum during their lifetime (between start and finish). With an average project length of just under two years (range = 6 to 72 months), this means there was, on average, about 4.75 months between inflection points.

Factors Contributing to Momentum Increases and Decreases

Project managers who are aware of the events and activities that positively and negatively influence momentum, and the effect these events may have on various constituents, can better manage momentum, and thus potentially shorten implementation time and improve the chances of project success. Table 1 lists the top ten factors contributing to momentum increases and decreases that were uncovered through the 51 retrospectives. These factors indicate the key levers available to project managers for maintaining and increasing momentum during the course of an IT project. Note that some factors, such as a change in project leadership, could, in different circumstances, contribute to either an increase or decrease in momentum.

Momentum Increasers	Momentum Decreasers
1. Perceived progress toward goal	1. Slow progress and/or missed deadlines
2. Launch events tied to project	2. Resource constraints
3. Communication; e.g., key announcements	3. Technical problems
4. Change in project leadership	4. Requirements issues
5. Sponsor encouragement	5. Ineffective/changing project leadership
6. Adding resources	6. Contractor issues
7. Social interaction; e.g., team bonding	7. Low morale
8. Sense of big picture impact	8. Lack of communication
9. Training	9. Competing projects
10. Stabilizing the environment	10. Holidays

Table 1. Factors Contributing to Momentum Increases and Decreases

EVALUATING PROJECT SUCCESS

Based on a review of the literature and experience gleaned through conducting retrospectives over the past 10 years, evaluating project success should include both *process* and *outcome* criteria, and ultimately focus on overall, weighted stakeholder satisfaction as illustrated in Figure 2.

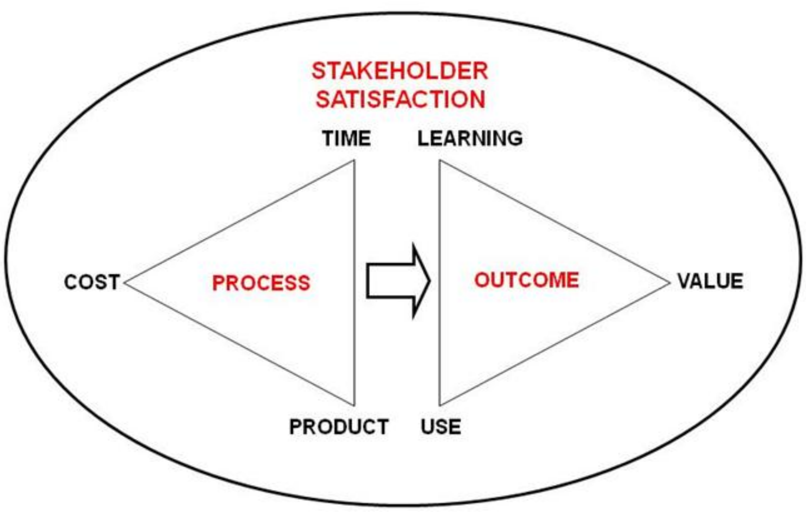

Figure 2. Project Success Criteria (Nelson 2005)

The three *process-related* criteria[2] include:

1. Time – The project came in on schedule.
2. Cost – The project came in on budget.
3. Product – The project produced a product of acceptable quality, and met other product-related specifications, including requirements, usability, ease of use, modifiability, and maintainability.

The three *outcome-related* criteria include:

4. Use – The project's resulting product/service is being used by its target constituencies (Pinto, and Slevin, 1988).
5. Learning – The project increased stakeholder knowledge and helped prepare the organization for future challenges (Shenhar, Dvir, Levy, and Maltz, 2001).
6. Value – The project will directly result in improved efficiency and/or effectiveness for the client organization(s). Common metrics include NPV, IRR, EVA, and the balanced scorecard (Pinto and Slevin, 1988).

Managing Stakeholder Perspectives

Project managers need to pay careful attention to the absolute and relative importance of each success criterion for each stakeholder group, while managing the inherent tradeoffs among the various criteria. The findings suggest that project managers and team members tend to be more process-centric, focusing primarily on bringing a project in on time and meeting specifications (Nelson, 2005). In contrast, project sponsors and top management are most concerned about a project's value to the organization. Not surprisingly, users care most about whether or not the project's outcome (e.g., system) would be used by its intended clients. In the 2005 study, none of the groups ranked learning (preparing for the future) in their top three criteria; although all of them suggested that learning was of at least moderate importance. The top three success criteria for all stakeholder groups were product, use and value (in descending order of importance). Project cost was ranked lowest overall.

Failed Successes and Successful Failures

Another interesting perspective on evaluating project success emerges in the form of "failed successes" (process success + outcome failure), and "successful failures" (process failure + outcome success). In other words, some projects are seen as successful from the process perspective (i.e., they meet specifications and come in on schedule and on budget), but are seen as failures from the outcome perspective (i.e., they don't add enough net value to the organization). An underlying theme in these failed successes is the importance of strategic and business process alignment. In many cases, successful efforts are derailed by changes in the macro environment.

On the other hand, a project shouldn't be written off as a failure just because it doesn't navigate the development process well. Projects that cost more and take longer to complete, yet deliver solutions that are used to solve business problems and, as a result, end up adding net value to the organization have become a very common scenario in the world of IT project management.

LESSONS LEARNED: CLASSIC MISTAKES AND BEST PRACTICES

> *"Some ineffective [project management] practices have been chosen so often, by so many people, with such predictable, bad results that they deserve to be called 'classic mistakes."* ~ Steve McConnell (1996)

After studying project success/failure as described above, it becomes apparent that failure is seldom a result of chance. Instead, it is rooted in one, or a series of, missteps by project managers. As McConnell (1996) suggests, we tend to make some mistakes more often than others, and he groups three dozen classic mistakes into the four categories of people, process, product, and technology.

Project managers need to closely examine past mistakes such as these, understand which are more common than others, and search for patterns that might help them avoid repeating the same mistakes in the future. To this end, the following is a description of the lessons learned from 130 IT projects.

---

[2] These three criteria are often referred to as the "triple constraints," or "trade-off triangle;" see for example: *Chaos Chronicles*, 2003; DeBrabander and Edstrom, 1977; Sauer, 1993).

Classic Mistakes (descending order of occurrence)	Category	No. of Projects	% of Projects
1. Poor estimation and/or scheduling	Process	83	67%
2. Insufficient risk management	Process	74	60%
3. Ineffective stakeholder management	People	65	52%
4. Insufficient planning	Process	62	50%
5. Shortchanged quality assurance	Process	55	44%
6. Weak personnel and/or team issues	People	53	43%
7. Poor requirements determination	Process	51	41%
8. Scope creep	Product	50	40%
9. Insufficient project sponsorship	People	43	35%
10. Unrealistic expectations	People	42	34%
11. Lack of user involvement	People	41	33%
12. Contractor failure	Process	36	29%
13. Inattention to politics	People	35	28%
14. Insufficient management controls	Process	33	27%
15. Undermined motivation	People	30	24%
16. Wishful thinking	People	28	23%
17. Wasted time in the fuzzy front-end	Process	25	20%
18. Code-like-hell programming	Process	24	19%
19. Research-oriented development	Product	24	19%
20. Friction between developers & customers	People	23	19%
21. Inadequate design	Process	23	19%
22. Abandonment of planning under pressure	Process	20	16%
23. Heroics	People	20	16%
24. Silver-bullet syndrome	Technology	17	14%
25. Planning to catch up later	Process	15	12%
26. Adding people to a late project	People	13	10%
27. Switching tools in the middle of a project	Technology	13	10%
28. Lack of automated source-code control	Technology	12	10%
29. Overestimated savings from new tools or methods	Technology	12	10%
30. Requirements gold-plating	Product	12	10%
31. Developer gold-plating	Product	8	6%
32. Insufficient resources	Process	8	6%
33. Push-me, pull-me negotiation	Product	6	5%
34. Uncontrolled problem employees	People	5	4%
35. Premature or overly frequent convergence	Process	4	3%
36. Noisy, crowded offices	People	3	2%

Table 2. Ranking of Classic Mistakes

The first major finding was that the vast majority of the classic mistakes were categorized as either process mistakes (48%) or people mistakes (38%); see Table 2. The remaining 14% were categorized as either product mistakes (9%) or technology mistakes (5%). None of the top ten mistakes was a technology mistake, which confirms that technology is seldom the chief cause of project failure. Therefore, technological expertise will rarely be enough to bring a project in on-time, on-schedule, while meeting requirements. Instead, this finding suggests that project managers should be, first and foremost, experts in managing processes and people.

It is important to note that the top four mistakes occurred in at least half of the projects examined. This finding clearly shows that if the project managers in the projects studied had focused their attention on better estimation and scheduling, risk management, stakeholder management, and planning they could have significantly improved their likelihood of success.

Based on these findings, project management offices (PMOs) would be wise to focus their education and training efforts first in these areas, while simultaneously instituting best practices that address these shortcomings within their organizational context. When instituting these best practices, it is best to cross-reference them with the classic mistakes. The matrix in Table 3 does just that. It matches some frequently cited best practices from our retrospectives with the top ten classic mistakes.

	Classic Mistakes	Agile Development	Communication Plan	Estimate-Convergence Graph	Joint Application Development (JAD)	Comprehensive Project Charter	Project Management Office	Retrospectives	Staged Delivery	Stakeholder Assessment	Work Breakdown Structure
1	Poor estimation and/or scheduling	X		X		X	X	X	X		X
2	Insufficient risk management			X		X	X	X	X		
3	Ineffective stakeholder management		X		X	X	X			X	
4	Insufficient planning			X		X	X	X			X
5	Shortchanged quality assurance	X			X				X		
6	Weak personnel and/or team issues	X	X				X	X	X		
7	Poor requirements determination	X			X						X
8	Scope Creep			X							X
9	Insufficient project sponsorship		X		X	X	X			X	
10	Unrealistic expectations	X	X	X	X	X				X	

Table 3. Classic Mistakes and Best Practices Matrix

On the front-end of a project, project managers and PMOs are encouraged to proactively identify the problems most likely to arise in each project and then use the matrix to help prioritize their project-specific best practices. For example, whereas all projects will want to use best practices to address the top four classic mistakes, a politically charged project would likely benefit the most from a well-thought-out stakeholder assessment and communication plan. As a second example, projects with ill-defined requirements or where heavy user involvement is necessary, a number of our retrospectives recommended agile development.

ROOT CAUSE ANALYSIS

Root cause analysis (RCA) is a tool designed to help identify not only what and how a problem or event occurred, but also why it happened. The practice of RCA is predicated on the belief that problems are best solved by attempting to correct or eliminate root causes, as opposed to merely addressing the obvious symptoms (e.g., classic mistakes). By directing corrective measures at root causes, it is hoped that the likelihood of problem recurrence will be minimized.

Systems-based RCA has emerged as an amalgamation of several "schools" of RCA, including safety-based RCA (e.g., accident investigation), production-based RCA (e.g., quality control for industrial engineering), and process-based RCA (e.g., the practice of business process improvement), along with ideas taken from fields such as change management, risk management, and systems thinking.

Despite the seeming disparity in purpose and definition among the various implementations of root cause analysis, there are some general principles that could be considered as universal. Similarly, it is possible to define a general process for performing RCA.

General Principles of Root Cause Analysis[3]

1. Directing corrective measures at root causes is more effective than merely treating the symptoms of a problem.
2. To be effective, RCA must be performed systematically, and conclusions must be backed up by evidence.
3. There is usually more than one root cause for any given problem; events are usually the result of a combination of contributors.
4. There is value in reducing the number of root causes to a manageable (actionable) set.
5. To be effective, root causes should be as specific as possible. For example, identifying “severe weather” as the root cause of data loss is not appropriate. Severe weather is not controlled by management. Similarly, if analysts arrive at vague recommendations such as, “Improve adherence to written policies and procedures,” then they probably have not found a basic and specific enough cause and need to expend more effort in the analysis process.

General Process for Performing Root Cause Analysis

1. Define/scope the problem; e.g., behind schedule, over budget, doesn’t meet requirements, not used, failure to learn, and/or doesn’t add business value.
2. Gather data/evidence; e.g., interviews, artifacts, and a project retrospective.
3. Identify causal factors and analyze cause/effect relationships; tools include fishbone diagram and causal factor chart.
4. Identify root causes; tools include root cause maps and “the five whys.”
5. Generate and implement recommendations; goal should be achievable recommendations for preventing reoccurrence, tracked to completion; tools include root cause summary table.

RCA Case Example

To illustrate the value of RCA in practice, a case study of a non-profit organization’s attempt to upgrade their Great Plains ERP software is provided. The upgrade to GP 8.0 occurred during the fourth quarter of 2005. A retrospective was conducted by an outside facilitator (the author) in January 2006. Twelve project stakeholders participated in the review, including the project manager, team members, representatives from all user groups (Finance and Accounting), IT support (help desk, systems architect, and network engineer), upper management (CIO and CFO), and an outside consulting firm that helped with the upgrade. The final report followed the five steps of RCA described above.

First, an evaluation of success concluded that the project could be best classified as a “successful failure.” That is, while the project came in over budget and late, with a list of 50 “unresolved issues,” the upgraded GP software was operational and in use, and key stakeholders were optimistic that the intended positive impact on the business would be realized (e.g., faster processing of accounts payable, better financial reporting, etc.).

Second, based on retrospective interviews and artifact analysis a cause-effect (a.k.a. fishbone) diagram was developed with the objective of isolating root causes. As depicted in Figure 3, four main categories of causes (people, process, product and technology) were determined to have contributed to the project being over budget, late and with unresolved issues. Next, a list of eight causal factors was compiled, and each causal factor was attached as a sub-branch to the diagram. Consistent with most projects, the vast majority of factors were found to be people or process-related.

A series of why questions, helped to further identify increasingly detailed levels of causes. For example,

Q: Why did insufficient planning occur on this project?

A: Misalignment of project objectives.

> Q: Why did the misalignment occur?
>
> A: IT personnel were focused on a “plain vanilla” solution, while users wanted a custom solution.
>
> > Q: What caused this to occur?
> >
> > A: Project manager inexperience; he failed to initiate a complete stakeholder and requirements analysis.

---

[3] Adapted from http://en.wikipedia.org/wiki/Root_cause_analysis (accessed 3/1/2010). Also see Bjørnson et al., 2009.

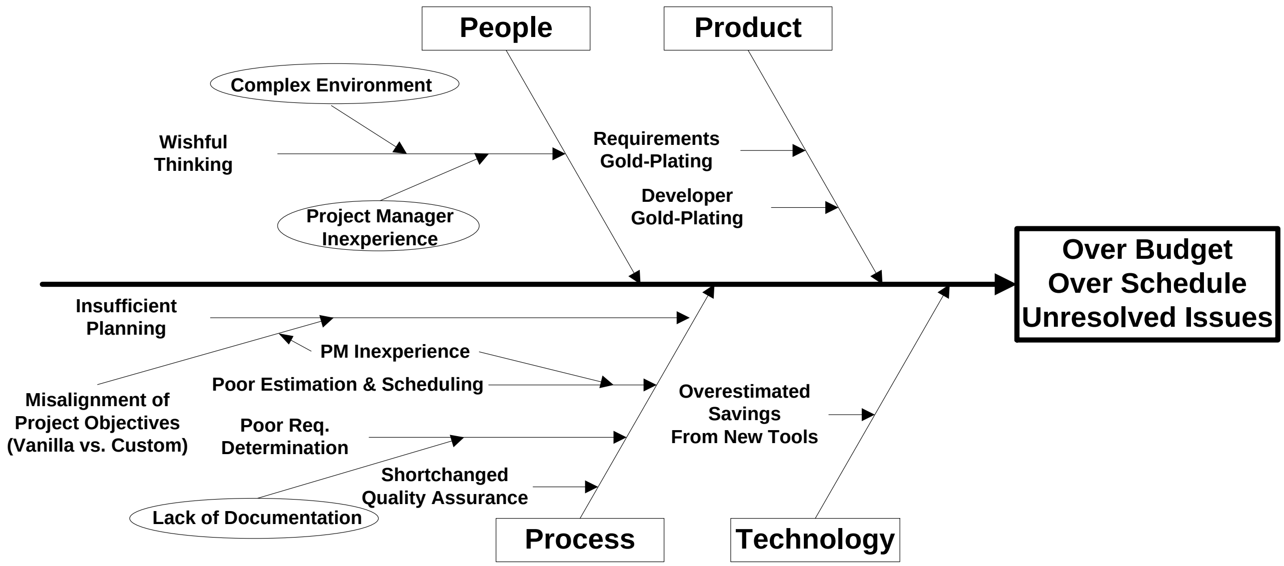

Figure 3. Cause and Effect Diagram

The RCA revealed that there were three root causes of the project being late, over budget, and with unresolved issues: project manager inexperience combined with an overly complex environment and poor documentation of existing systems. These root causes are circled on the cause-effect diagram indicating that they are actionable and should be addressed in order to prevent similar results from occurring in future projects.

Of course, RCA is also useful in uncovering positive effects. In this case, based on the retrospective interviews, it was concluded that two things had gone particularly well during the course of the project:

- Outsourcing/grafting of experienced personnel. The consultants brought in to assist with the project had a tremendous knowledge of the Great Plains application. This proved to be invaluable when product customization was requested by the end users.
- Stakeholder management. Both the consultants and IT project team members did a good job of managing key stakeholders throughout the project. As one end user commented, "The consultants are hardworking, professional and available." Another stated that the "[project manager] is very supportive and attentive."

It was suggested that without these two positive effects, the project would have likely been an outright failure.

RECOMMENDATIONS

As stated earlier, in many cases retrospectives are seen as not adding value because they conclude with a laundry list of un-prioritized recommendations. For a retrospective to have any impact on future behaviors, they need to have a carefully constructed recommendations section that answers the questions of WHAT? (specifically what needs to be done); WHY? (rationale behind the recommendation); WHO? (the person(s) responsible for acting on the recommendation); and WHEN? (typically presented as now (< 3 months), soon (3-6 months), or later (6-12 months)).

Following through with the non-profit example described above, the following table outlines three actionable recommendations:

What and Why	Who	When
1. Project management training and a mentorship program for IT personnel involved in project management activities. Rationale – see RCA	Project Management Office & Human Resources department	**Now** (<3 months)
2. Increased documentation of existing information systems. Rationale – see RCA	Chief Technology Architect	**Soon** (3-6 months)
3. A new policy promoting simplification, standardization, and integration of information systems across the organization. Rationale – see RCA	Chief Information Officer & IT Steering Committee	**Later** (6-12 months)

Table 4. Example of an Actionable Set of Recommendations

CONCLUSION

Given the critical role that project management plays in the field of information technology, we need to accelerate our progress on this typically slow and painful learning process. To this end, project retrospectives need to evolve beyond simple checklists of what went right and wrong to become more analytic, as exemplified by the process of action research described in this paper. Managers need to recognize that virtually every project experiences some successes and some failures. Yet, regardless of the level of success or failure, every project should contribute to organizational learning and continuous improvement.

Based on the analysis of 130 retrospectives between 1999 and 2009, a tremendous amount of learning can take place if (a) retrospectives are done effectively (as outlined in this paper), and (b) they are properly communicated to all relevant stakeholders. While this paper was focused on the former, it is suggested that future research and practice would benefit most by exploring best practices for promoting the transference of retrospective knowledge to future projects. As Oscar Wilde said,

"We teach people how to remember, we never teach them how to grow."

REFERENCES

1. Boddie, J. (1987) The Project Postmortem, *Computerworld*, 21, 49, 77-82.
2. Bjørnson, F., Wang, I., and Arisholm, E. (2009) Improving the effectiveness of root cause analysis in post mortem analysis: A controlled experiment, *Information and Software Technology*, 51, 1; 150.
3. *Chaos Chronicles*, *Version 3.0* (2003) The Standish Group International, West Yarmouth, MA.
4. Collier, B., DeMarco, T. and Fearey. P. (1996) A Defined Process for Project Postmortem Review, *IEEE Software*, 5-71.
5. DeBrabander, B. and Edstrom, A. (1977) Successful Information System Development Projects, *Management Science*, 24, 2, 191.
6. Derby, E. and Schwaber, K. (2006) Agile Retrospectives: Making Good Teams Great, Pragmatic Bookshelf.
7. Favaro, J. (2010) Renewing the Software Project Management Life Cycle, *IEEE Software*. 27, 1, 17.
8. Glass, R., (2002) Project Retrospectives, and Why They Never Happen, *IEEE Software*, 111-112.
9. Hayes, D. (2008) The Efficacy of Project Retrospectives on Organizational Process Improvement Initiatives in Software Development, Unpublished Dissertation, School of Computer Science & Information Systems, Pace University, 260 pgs.
10. Hoffman, T. (2005) After the fact: How to find out if your IT project made the grade, *Computerworld*.
11. Kasi, V., Keil, M., Mathiassen, L., and Pedersen, K. (2008) The post mortem paradox: a Delphi study of IT specialist perceptions, *European Journal of Information Systems*, 17, 1; 62-78.
12. Kerth, N. L. (2001) Project Retrospectives: A Handbook for Team Reviews, Dorset House Publishing, New York.
13. Lewin, K. (1946) Action research and minority problems, *Journal of Social Issues*, 2, 4, 34-46.
14. McConnell, S., (1996) Rapid Development, Microsoft Press, Redmund.
15. Nelson, R. R. and Jansen, K. J., (2009) Mapping and Managing Momentum in IT Projects, *MIS Quarterly Executive*, 8, 3, 141-148.

16. Nelson, R. R., (2007) IT Project Management: Infamous Failures, Classic Mistakes, and Best Practices, *MIS Quarterly Executive*, 6, 2, 67-78.
17. Nelson, R. R., (2005) Project Retrospectives: Evaluating Project Success, Failure, and Everything in Between, *MIS Quarterly Executive*, 4, 3, 361-372.
18. Pinto, J.K. and Slevin, D.P. (1988) Project Success: Definitions and Measurement Techniques, *Project Management Journal*, 19, 1, 67.
19. Reason, P. and Bradbury, H. (2007) Handbook of Action Research, 2nd Edition, Sage, London.
20. Sauer, C. (1993) Why Information Systems Fail: A Case Study Approach, Oxfordshire, Henley-On-Thames.
21. Shenhar, A.J., Dvir, D., Levy, O. and Maltz, A.C. (2001) Project Success: A Multidimensional Strategic Concept, *Long Range Planning*, 34, 6, 699.