IT Project Management: Lessons Learned from Project Retrospectives 1999–2020

Other titles in Foundations and Trends® in Information Systems

Structured Conflict Approaches used in Strategic Decision Making: from Mason's Initial Study to Virtual Teams
Jerry Fjermestad
ISBN: 978-1-68083-616-5

Qualitative and Critical Research in Information Systems and Human-Computer Interaction: Divergent and Convergent Paths
Eleanor Wynn and Helena Vallo Hult
ISBN: 978-1-68083-556-4

Enterprise Personal Analytics: The Next Frontier in Individual Information Systems Research
T. Clohessy, T. Acton, E. Whelan and W. Golden
ISBN: 978-1-68083-460-4

Empirical Research in Information Systems: 2001–2015
S. Shuraida, H. Barki and A. Luong
ISBN: 978-1-68083-430-7

IT Project Management: Lessons Learned from Project Retrospectives 1999–2020

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

now
the essence of knowledge
Boston — Delft

Foundations and Trends® in Information Systems

Published, sold and distributed by:
now Publishers Inc.
PO Box 1024
Hanover, MA 02339
United States
Tel. +1-781-985-4510
www.nowpublishers.com
sales@nowpublishers.com

Outside North America:
now Publishers Inc.
PO Box 179
2600 AD Delft
The Netherlands
Tel. +31-6-51115274

The preferred citation for this publication is

R. Ryan Nelson. *IT Project Management: Lessons Learned from Project Retrospectives 1999–2020*. Foundations and Trends® in Information Systems, vol. 4, no. 4, pp. 275–381, 2021.

ISBN: 978-1-68083-777-3

Foundations and Trends® in Information Systems
Volume 4, Issue 4, 2020

Editorial Board

Editorial Scope

Topics

Foundations and Trends® in Information Systems publishes survey and tutorial articles in the following topics:

- IS and Individuals
- IS and Groups
- IS and Organizations
- IS and Industries
- IS and Society
- IS Development
- IS Economics
- IS Management
- IS Research Methods

Information for Librarians

Foundations and Trends® in Information Systems, 2020, Volume 4, 4 issues. ISSN paper version 2331-1231. ISSN online version 2331-124X. Also available as a combined paper and online subscription.

Contents

IT Project Management: Lessons Learned from Project Retrospectives 1999–2020

R. Ryan Nelson

McIntire School of Commerce, University of Virginia, USA; rnelson@virginia.edu

ABSTRACT

Companies realize they cannot stay in business if they cannot manage their projects effectively. Yet, most organizations still are either unable or unwilling to perform the one basic activity critical to project management and continuous improvement: learning from mistakes and successes. This monograph provides a framework for conducting retrospectives—a process of "looking back"—to glean lessons for ongoing and future project success. This systematic approach has evolved through the analysis of hundreds of information technology (IT) projects over the past 20 years. Compiling the findings of this extensive research, the monograph offers a guide for how to leverage best practices to avoid classic mistakes with the end goal of improving the chances of project success. To this end, the monograph begins with a discussion on project retrospectives, including what they are, why they are important, and why they aren't done, followed by a description of the action research (i.e., meta-retrospective) on which the remainder of the monograph is based. The focus of Section 2 is on 10 of the most *infamous* IT project failures (each with reported losses of over $100 million), a ranked list of classic

R. Ryan Nelson (2021), "IT Project Management: Lessons Learned from Project Retrospectives 1999–2020", Foundations and Trends® in Information Systems: Vol. 4, No. 4, pp 275–381. DOI: 10.1561/2900000022.

mistakes that most often contribute to project failure (categorized by people, process, product, and technology), and a discussion of one method (root cause analysis) and five best practices designed to prevent classic mistakes from occurring in the first place. Section 3 presents a robust framework for evaluating project success based three process-related criteria (schedule, cost, and product) and three outcome criteria (use, value, and learning). Section 4 defines *momentum* as it relates specifically to IT projects and discusses how managers can equip themselves with mapping and analysis tools to control the momentum of a project for best results. The focus of Section 5 is on the most cited reason for IT project failure—*poor estimation*. Using the findings from two research studies, the section provides recommendations to help project managers improve project estimation. In sum, this monograph plots a pathway to success for IT project managers by applying the voluminous findings from analysis of retrospectives done for 264 IT projects from 1999 to 2020. The result is a comprehensive guide that project managers may use to gauge progress at points throughout a project's life, map momentum, apply best practices to spot and prevent classic mistakes, conduct root-cause analysis, and devise actionable recommendations that will help their organization achieve project success.

Keywords: project management; project success; project failure; retrospectives; post-implementation reviews; classic mistakes; best practices; root cause analysis; momentum; estimation.

1

Project Retrospectives

> If one wants to define the future, they must study the past.
> – Confucius

Failure to learn from past mistakes and successes has consistently been a major obstacle to improving IT project management. As Boddie wrote in 1987:

> We talk about software engineering but reject one of the most basic engineering practices; identifying and learning from our mistakes. Errors made while building one system appear in the next one. What we need to remember is the attention given to failures in the more established branches of engineering. In software projects, as in bridge building, a successful effort can do little more than affirm that the tools and methods used were appropriate for the task. By the same token, failed projects need more than explanation or rationalization if they are to teach any lessons.

This monograph addresses this shortcoming by integrating, updating, and extending the research findings from four previous studies on IT

project retrospectives: Nelson (2005, 2007), Nelson and Jansen (2009), and Nelson and Morris (2014). The result is a "meta-retrospective" of 264 IT projects analyzed as part of a program of action research conducted between 1999 and 2020.

1.1 What is a Retrospective?

A retrospective is a formal method for evaluating project performance, extracting lessons learned, and making recommendations for the future. The word "retrospective" means looking back on, contemplating, or directed to the past. In the IT industry, retrospectives go by many names. One popular term is postmortem, from the Latin for "after death." In addition to the obvious negative connotations attached to this label, IT projects don't, or at least aren't supposed to, end with death; rather they should bring something to life. As an alternative, the Latin term postpartum, meaning "after birth," is sometimes used, again with its own set of negative associations. Various branches of the military use their own terms: After Action Review or Post Engagement Redress (Army), Navy Lessons Learned or Hot Wash Up, and C-GULL (Coast Guard Uniform Lessons Learned). While each term has its following, the terms retrospective and post-implementation review seem to be the most descriptive without implying success or failure.

Another argument for using the term retrospective is that it is not limited to the post-implementation phase of a project. In fact, retrospectives conducted following Agile sprints[1] or critical milestones in a project's life cycle can either confirm that the project is on track or suggest mid-course adjustments before it's too late. In some cases, an interim retrospective may conclude that a project should be terminated, avoiding the dreaded "runaway" label that stigmatizes projects as out of control.

[1] An Agile sprint is a repeatable fixed time-box (typically two weeks) during which a product of the highest possible value is created.

1.2 Why Retrospectives Are Important

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

- **Organizational learning** – Get the collective story out (synergistic learning) and ensuring that individual stakeholders hear the whole story, not just their personal experience.
- **Continuous improvement** – Facilitate 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 at a minimum to break even (Kerth, 2001)—e.g., a one-day retrospective will save at least one day on the next project, and many more when one considers the potential impact across multiple projects.

1.3 Why Retrospectives Aren't Done

Regardless of what they're called, and despite their potential to yield significant benefits, retrospectives are rarely conducted outside of the military. According to the Gartner Group, an IT consulting firm, only 13% of their clients conduct such reviews (Hoffman, 2005). The most obvious reason is the natural desire to put the past to rest and move on to something new.

Organizations are also reluctant to allocate additional time and money to a project after the system is completed. 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. 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 only as "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, a trend that recently received attention in the Project Management Institute's (PMI) Body of Knowledge and the Software Engineering Institute's (SEI) Capability Maturity Model.

1.4 Retrospective Templates

Project retrospectives are scalable to fit virtually any project—regardless of size, scope, technology, methodology, industry, or geography. To help scale retrospectives to fit a specific need there are a number of different templates that can be applied—as exemplified in the appendix: IT Project Retrospective Templates. Template 1 is designed to capture a great deal of information on a project, and as a result is especially useful for medium to large scale retrospectives. Templates 2–5 are designed to be much leaner/lightweight in their approach and therefore a better fit for Agile projects (e.g., at the end of each two-week sprint). These smaller retrospectives can be done in less than 10 minutes with minimal documentation. Templates for conducting quick retrospectives can be found within online collaborative whiteboarding tools like Miro and Mural.

1.5 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." The process uses data-driven collaborative analysis to understand underlying causes, that enable future predictions about personal and organizational change (Lewin, 1946; Reason and Bradbury,

2007). This form of research is particularly well suited for assessing completed sprints/milestones/projects and making recommendations going forward.

Since the summer of 1999, the University of Virginia has offered a Master of Science in the Management of Information Technology (MS MIT) degree program in an executive format to working professionals.[2] To date, more than 2,000 people have participated in the program, each with an average of over 10 years of experience and direct involvement with at least one major IT project. All program participants receive instruction in how to conduct action research in the form of project retrospectives using the following framework (note: a more detailed template can be found in appendix: IT Project Retrospective Templates —Template 1: Medium to Large Scale Retrospectives):

> *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 (artifact analysis, stakeholder interviews, etc.)
>
> *Project Timeline and Momentum Map*
>
> *Evaluation of Project Success/Failure*
>
> *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, participants work in teams and conduct retrospectives of recently completed IT projects. Thus far, 264 retrospectives have been conducted in 192 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.

[2]For more information on the MS in the Management of Information Technology (MS MIT) program at the University of Virginia visit: https://www.commerce.virginia.edu/ms-mit.

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 particular timeframe. When viewed as a whole, these 264 projects provide an incredible opportunity to understand project management practices at a more macro level and to 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). These findings will be discussed in detail in Section 3.

The study reported in Nelson (2007) focused on the lessons-learned portion of each retrospective (regardless of whether the project was ultimately considered a success). The analysis yielded intriguing findings on what tended to go wrong with the 99 projects studied through 2006.

In 2009, Nelson and Jansen reported on the results of mapping the momentum of 51 projects. This study presented both retrospective observations as well as guidelines for helping manage momentum through the course of a project. Section 4 will review the highlights of this study.

In 2014, Nelson and Morris presented the findings from two survey-based studies on IT project estimation, which replicated and then extended a study conducted 20 years earlier. Guidelines are provided for improving project estimation, taking account of the greater use today of Agile, rather than traditional Waterfall, development methods. This study will be summarized in Section 5.

Together, these reports provide a longitudinal meta-retrospective—a deep study across a long time span—with synergistic qualities.

References

Adler, P. (1981). *Momentum: A Theory of Social Action*. SAGE.

Barrett, L. and S. Gallagher (2003). "Fast food fails digital networking test". *Baseline*. URL: http://www.baselinemag.com/article2/0,3959,1173624,00.asp.

Boddie, J. (1987). "The project postmortem". *Computerworld*. 21(49): 77–82.

Boehm, B. (1981a). "An experiment in small-scale application software engineering". *IEEE Transactions on Software Engineering*. SE7(5): 482–494.

Boehm, B. (1981b). *Software Engineering Economics*. Prentice Hall.

Brill, S. (2016). "Is America any safer?" *The Atlantic*. URL: https://www.theatlantic.com/magazine/archive/2016/09/are-we-any-safer/492761/.

Brooks, F. (1975). *The Mythical Man-Month*. Addison-Wesley.

Buehler, R., D. Griffin, and M. Ross (2002). "Inside the planning fallacy: The causes and consequences of optimistic time predictions". In: *Heuristics and Biases: The Psychology of Intuitive Judgment*. Ed. by T. Gilovich, D. Griffin, and D. Kahneman. Cambridge University Press. 250–270.

Burke, K. L., M. M. Burke, and A. B. Joyner (1999). "Perceptions of momentum in college and high school basketball: An exploratory case study investigation". *Journal of Sport Behavior*. (22): 303–309.

Burton-Jones, A., A. Gilchrist, P. Green, and M. Draheim (2020). "Improving social alignment during digital transformation". *Communications of the ACM*. 63(9): 65–71.

Charette, R. (2013). "The U.S. air force explains its $1 billion ECSS bonfire". *IEEE Spectrum*. URL: http://spectrum.ieee.org/riskfactor/aerospace/military/the-us-air-force-explains-its-billion-ecss-bonfire.

Charette, R. (2018). "The biggest IT failures of 2018". *IEEE Spectrum*. 27 Dec. 2018.

Cohn, M. (2005). *Agile Estimating and Planning*. Prentice Hall.

Cone, E. (2002). "The ugly history of tool development at the FAA". *Baseline*. URL: http://www.baselinemag.com/article2/0,1540,794112,00.asp.

DeMarco, T. (1982). *Controlling Software Projects: Management, Measurement and Estimation*. Yourdon Press.

DeMarco, T. and T. Lister (1999). *Peopleware: Productive Projects and Teams*. 2nd ed. Dorset House Publishing.

DeMarco, T. and T. Lister (2013). *Peopleware: Productive Projects and Teams*. 3rd ed. Addison-Wesley.

Derby, E. and D. Schwaber (2006). *Agile Retrospectives: Making Good Teams Great*. Pragmatic Bookshelf.

Dutton, J. E. and R. B. Duncan (1987). "The creation of momentum for change through the process of strategic issue diagnosis". *Strategic Management Journal*. 8(3): 279–295.

Eggen, D. (2005). "FBI pushed ahead with troubled software". *Washington Post*. URL: http://www.washingtonpost.com/wp-dyn/content/article/2005/06/05/AR2005060501213.html.

Fruhlinger, J., T. Wailgum, and P. Sayer (2020). "16 famous ERP disasters, dustups and disappointments". *CIO*. 20 Mar. 2020.

Futrell, R. T., D. F. Shafer, and L. I. Shafer (2002). *Quality Software Project Management*. Prentice Hall.

Gersick, C. J. G. (1994). "Pacing strategic change: The case of a new venture". *Academy of Management Journal*. 37(1): 9–45.

Ginsberg, A. and N. Venkatraman (1995). "Institutional initiatives for technological change: From issue interpretation to strategic choice". *Organization Studies*. 16(3): 425–448.

Glass, R. (1997). *Software Runaways: Lessons Learned from Massive Software Project Failures*. Prentice Hall.

Herper, M. (2017). "MD Anderson benches IBM Watson in setback for artificial intelligence in medicine". *Forbes*. URL: https://www.forbes.com/sites/matthewherper/2017/02/19/md-anderson-benches-ibm-watson-in-setback-for-artificial-intelligence-in-medicine/#306ecec63774.

Hoffman, T. (2005). "After the fact: How to find out if your IT project made the grade". *Computerworld*. 11 Jul. 2005.

Jansen, K. J. (2004). "From persistence to pursuit: A longitudinal examination of momentum during the early stages of strategic change". *Organization Science*. 15(3): 276–294.

Jones, C. (1994). *Assessment and Control of Software Risks*. Yourdon Press.

Jones, C. (1998). *Estimating Software Costs*. McGraw-Hill.

Kahneman, D. and A. A. Tversky (1979). "Intuitive prediction: Biases and corrective procedures". *TIMS Studies in Management Science*. 12: 313–327.

Kalin, S. (2002). "The ROI of application integration". *CIO*. URL: http://www.cio.com/article/2440628/enterprise-software/the-roi-of-application-integration.html.

Keil, M. (1995). "Pulling the plug: Software project management and the problem of project escalation". *MIS Quarterly*. 19(4): 421–447.

Kerth, N. L. (2001). *Project Retrospectives: A Handbook for Team Reviews*. Dorset House Publishing.

Khurana, A. and S. R. Rosenthal (1997). "Integrating the fuzzy front end of new product development". *Sloan Management Review*. 38(2): 103–120.

Lakhanpal, B. (1993). "Understanding the factors influencing the performance of software development groups: An exploratory group-level analysis". *Information and Software Technology*. 35(8): 468–473.

Larson, C. and F. LaFasto (1989). *Teamwork: What Must Go Right, What Can Go Wrong*. SAGE.

Lederer, A. L. and J. Prasad (1992). "Nine management guidelines for better cost estimating". *Communications of the ACM*. 35(2): 51–59.

Lewin, K. (1946). "Action research and minority problems". *Journal of Social Issues*. 2(4): 34–46.

Lyytinen, K. and R. Hirschheim (1987). "Information systems failures: A survey and classification of the empirical literature". *Oxford Surveys in Information Technology*. 4: 257–309.

McConnell, S. (1996). *Rapid Development*. Microsoft Press.

McConnell, S. (2006). *Software Estimation: Demystifying the Black Art*. Microsoft Press.

McDougall, P. (2006). "8 expensive IT blunders". *InformationWeek*. 16(October).

Melik, R. (2007). *The Rise of the Project Workforce: Managing People and Projects in a Flat World*. Wiley.

Montealegre, R., H. Nelson, C. Knoop, and L. Applegate (1996). "BAE automated systems (A): Denver international airport baggage-handling system". *Harvard Business School Case* #9-396-311.

Nelson, R. R. (2005). "Project retrospectives: Evaluating project success, failure, and everything in between". *MIS Quarterly Executive*. 4(3): 361–372.

Nelson, R. R. (2007). "IT project management: Infamous failures, classic mistakes, and best practices". *MIS Quarterly Executive*. 6(2): 67–78.

Nelson, R. R. and K. J. Jansen (2009). "Mapping and managing momentum in IT projects". *MIS Quarterly Executive*. 8(3): 141–148.

Nelson, R. R. and M. Morris (2014). "IT project estimation: Contemporary practices and management guidelines". *MIS Quarterly Executive*. 13(1): 15–30.

Perez, S. (2018). "Looking back at Google+". *Techcrunch*. URL: https://techcrunch.com/2018/10/08/looking-back-at-google/.

Pezzo, M., J. Litman, and S. Pezzo (2006). "On the distinction between yuppies and hippies: Individual differences in prediction biases for planning future tasks". *Personality and Individual Differences*. 41(7): 1359–1371.

Pinto, J. K. and D. P. Slevin (1988). "Project success: Definitions and measurement techniques". *Project Management Journal*. 19(1): 67.

Reason, P. and H. Bradbury (2007). *Handbook of Action Research*. 2nd ed. SAGE.

Rooney, J. and L. Vanden Heuvel (2004). "Root cause analysis for beginners". *Quality Progress*. 37(7): 45–53.

Sabherwal, R., R. Hirschheim, and T. Goles (2001). "The dynamics of alignment: Insights from a punctuated equilibrium model". *Organization Science*. 12(2): 179–197.

Sackman, H., W. J. Erikson, and E. E. Grant (1968). "Exploratory experimental studies comparing online and offline programming performance". *Communications of the ACM*. 1(11): 3–11.

Sauer, C. (1993). *Why Information Systems Fail: A Case Study Approach*. Alfred Waller Limited.

Shenhar, A. J., D. Dvir, O. Levy, and A. C. Maltz (2001). "Project success: A multidimensional strategic concept". *Long Range Planning*. 34(6): 699.

Thomsett, R. (2002). *Radical Project Management*. Prentice Hall.

Tufte, E. R. (1983). *The Visual Display of Quantitative Information*. Graphics Press. 9.

Tyre, M. J. and W. J. Orlikowski (1994). "Windows of opportunity: Temporal patterns of technological adaptation in organizations". *Organization Science*. 5(1): 98–118.

Varon, E. (2004). "No easy IT fix for the IRS". *CIO*. URL: http://www.cio.com/article/2439729/outsourcing/no-easy-it-fix-for-the-irs.html.

Willcocks, L. and M. Lacity (2006). *Global Sourcing of Business and IT Services*. Palgrave Macmillan.

Wilson, T., T. Wheatley, J. Meyers, D. Gilbert, and D. Axsom (2000). "Focalism: A source of durability bias in affective forecasting". *Journal of Personality and Social Psychology*. 78(5): 821–836.

Yourdon, E. (1999). *Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects*. Upper Saddle River, NJ: Prentice Hall.