MIS Quarterly Executive

IT Project Management: Infamous Failures, Classic Mistakes, and Best Practices[1]

R. Ryan Nelson
University of Virginia

Executive Summary

In recent years, IT project failures have received a great deal of attention in the press as well as the boardroom. In an attempt to avoid disasters going forward, many organizations are now learning from the past by conducting retrospectives—that is, project postmortems or post-implementation reviews. While each individual retrospective tells a unique story and contributes to organizational learning, even more insight can be gained by examining multiple retrospectives across a variety of organizations over time. This research aggregates the knowledge gained from 99 retrospectives conducted in 74 organizations over the past seven years. It uses the findings to reveal the most common mistakes and suggest best practices for more effective project management.[2]

INFAMOUS FAILURES

"Insanity: doing the same thing over and over again and expecting different results."
— Albert Einstein

If failure teaches more than success, and if we are to believe the frequently quoted statistic that two out of three IT projects fail,[3] then the IT profession must be developing an army of brilliant project managers. Yet, although some project managers are undoubtedly learning from experience, the failure rate does not seem to be decreasing. This lack of statistical improvement may be due to the rising size and complexity of projects, the increasing dispersion of development teams, and the reluctance of many organizations to perform project retrospectives.[4] There continues to be a seemingly endless stream of spectacular IT project failures. No wonder managers want to know what went wrong in the hopes that they can avoid similar outcomes going forward.

Figure 1 contains brief descriptions of 10 of the most *infamous* IT project failures. These 10 represent the tip of the iceberg. They were chosen because they are the most heavily cited[5] and because their magnitude is so large. Each one reported losses over $100 million. Other than size, these projects seem to have little in common. One-half come from the public sector, representing billions of dollars in wasted taxpayer dollars and lost services, and the other half come from the private sector, representing billions of dollars in added costs, lost revenues, and lost jobs.

1 Jeanne Ross was the Senior Editor, Keri Pearlson and Joseph Rottman were the Editorial Board Members for this article for this article.

2 The author would like to thank Jeanne Ross, anonymous members of the editorial board, Barbara McNurlin, and my colleague Barb Wixom, for their comments and suggestions for improving this article.

3 The Standish Group reports that roughly two out of three IT projects are considered to be failures (suffering from total failure, cost overruns, time overruns, or a rollout with fewer features or functions than promised).

4 Just 13% of the Gartner Group's clients conduct such reviews, says Joseph Stage, a consultant at the Stamford, Connecticut-based firm. Quoted in Hoffman, T. "After the Fact: How to Find Out If Your IT Project Made the Grade," *Computerworld*, July 11, 2005.

5 Glass, R. *Software Runaways: Lessons Learned from Massive Software Project Failures*, 1997; Yourdon, E. *Death March: The Complete Software Developer's Guide to Surviving 'Mission Impossible' Projects*, 1999. "To Hell and Back: CIOs Reveal the Projects That Did Not Kill Them and Made Them Stronger," *CIO Magazine*, December 1, 1998. "Top 10 Corporate Information Technology Failures," *Computerworld*: http://www.computerworld.com/computerworld/records/images/pdf/44NfailChart.pdf; McDougall, P. "8 Expensive IT Blunders," *InformationWeek*, October 16, 2006.

Figure 1: Infamous IT Project Failures

Internal Revenue Service — PROJECT: Business Systems Modernization; Launched in 1999 to upgrade the agency's IT infrastructure and more than 100 business applications.

WHAT HAPPENED? By assembling a star-studded team of vendors, the IRS thought its $8 billion modernization project would manage itself. The IRS thought wrong. As a result, the agency's ability to collect revenue, conduct audits, and go after tax evaders was severely compromised. This case study illustrates what can go wrong when a complex project overwhelms the management capabilities of both vendor and client. Some consider it to be the most expensive systems development "fiasco" in history, with delays costing the U.S. Treasury tens of billions of dollars per year.

Source: http://www.cio.com/archive/040104/irs.html.

Federal Aviation Administration — PROJECT: Advanced Automation System (AAS); FAA's effort to modernize the nation's air traffic control system.

WHAT HAPPENED? AAS was originally estimated to cost $2.5 billion with a completion date of 1996. The program, however, experienced numerous delays and cost overruns, which were blamed on both the FAA and the primary contractor, IBM. According to the General Accounting Office, almost $1.5 billion of the $2.6 billion spent on AAS was completely wasted. One participant remarked, "It may have been the greatest failure in the history of organized work."

Source: http://www.baselinemag.com/article2/0,1540,794112,00.asp.

Federal Bureau of Investigation — PROJECT: "Trilogy;" Four-year, $500M overhaul of the FBI's antiquated computer system.

WHAT HAPPENED? Requirements were ill-defined from the beginning and changed dramatically after 9/11 (agency mission switched from criminal to intelligence focus) thus creating a strained relationship between the FBI and its primary contractor, SAIC. As Senator Patrick Leahy (D-Vt.) stated, "This project has been a train wreck in slow motion, at a cost of $170M to American taxpayers and an unknown cost to public safety."

Sources: http://www.cio.com/archive/061505/gmen.html,
http://www.npr.org/templates/story/story.php?storyId=4283204,
http://www.washingtonpost.com/wp-dyn/content/article/2005/06/05/AR2005060501213.html.

McDonalds — PROJECT: "Innovate;" Digital network for creating a real-time enterprise.

WHAT HAPPENED? Conceived in January 2001, Innovate was the most expensive (planned to spend $1 billion over five years) and intensive IT project in company history. Eventually, executives in company headquarters would have been able to see how soda dispensers and frying machines in every store were performing, at any moment. After two years and $170M, the fast food giant threw in the towel.

Source: http://www.baselinemag.com/article2/0,3959,1173624,00.asp.

Denver International Airport — PROJECT: Baggage-handling system.

WHAT HAPPENED? It took 10 years and at least $600 million to figure out big muscles, not computers, can best move baggage. The baggage system, designed and built by BAE Automated Systems Inc., launched, chewed up, and spit out bags so often that it became known as the "baggage system from hell." In 1994 and 1995, the baggage system kept Denver's new airport from opening. When it finally did open, the baggage system didn't. It was such a colossal failure that every airline except United Airlines refused to use it. And now, finally and miserably, even United is throwing in the towel.

Sources: http://www.msnbc.msn.com/id/8975649/; BAE Automated Systems (A): Denver International Airport Baggage-Handling System, Harvard Business School Case #9-396-311, November 6, 1996;

http://www.computerworld.com/managementtopics/management/project/story/0,10801,102405,00.html?source=NLT_AM&nid=102405.

Figure 1: Infamous IT Project Failures	
AMR Corp., Budget Rent A Car Corp., Hilton Hotels Corp., Marriott International Inc.	PROJECT: "Confirm;" Reservation system for hotel and rental car bookings.
WHAT HAPPENED? After four years and $125 million in development, the project crumbled in 1992 when it became clear that Confirm would miss its deadline by as much as two years. AMR sued its three partners for breach of contract, citing mismanagement and fickle goals. Marriott countersued, accusing AMR of botching the project and covering it up. Both suits were later settled for undisclosed terms. Confirm died and AMR took a $109 million write-off. Sources: http://sunset.usc.edu/classes/cs510_2004/notes/confirm.pdf http://www.computerworld.com/computerworld/records/images/pdf/44NfailChart.pdf	
Bank of America	PROJECT: "MasterNet;" Trust accounting system.
WHAT HAPPENED? In February 1988, hardware problems caused the Bank of America (BofA) to lose control of several billion dollars of trust accounts. All the money was eventually found in the system, but all 255 people—i.e., the entire Trust Department—were fired, as all the depositors withdrew their money. This is a classic case study on the need for risk assessment, including people, process, and technology-related risk. BofA spent $60M to fix the $20M project before deciding to abandon it altogether. BofA fell from being the largest bank in the world to No. 29. Source: http://sunset.usc.edu/classes/cs510_2001/notes/masternet.pdf.	
KMart	PROJECT: IT systems modernization.
WHAT HAPPENED? To better compete with its rival, Wal-Mart Corp., in Bentonville, Ark., retailer Kmart Corp., in Troy, Mich., launched a $1.4 billion IT modernization effort in 2000 aimed at linking its sales, marketing, supply, and logistics systems. But Wal-Mart proved too formidable, and 18 months later, cash-strapped Kmart cut back on modernization, writing off the $130 million it had already invested in IT. Four months later, it declared bankruptcy. Source: http://www.spectrum.ieee.org/sep05/1685.	
London Stock Exchange	PROJECT: "Taurus;" Paperless share settlement system.
WHAT HAPPENED? In early 1993, the London Stock Exchange abandoned the development of Taurus after more than 10 years of development effort had been wasted. The Taurus project manager estimates that, when the project was abandoned, it had cost the City of London over £800 million. Its original budget was slightly above £6 million. Taurus was 11 years late and 13,200 percent over budget without any viable solution in sight. Source: http://www.it-cortex.com/Examples_f.htm.	
Nike	PROJECT: Integrated enterprise software.
WHAT HAPPENED? Nike spent $400 million to overhaul its supply chain infrastructure, installing ERP, CRM, and SCM—the full complement of analyst-blessed integrated enterprise software. Post-implementation (3rd quarter, 2000), the Beaverton, Ore.-based sneaker maker saw profits drop by $100 million, thanks, in part, to a major inventory glitch (it over-produced some shoe models and under-produced others). "This is what I get for our $400 million?" said CEO Phil Knight. Source: http://www.cio.com/archive/081502/roi.html.	

The postmortem in each case contains clues as to what went wrong. While some of the projects experienced contractor failure (IRS, FAA, FBI), others cited poor requirements determination (FAA, FBI), ineffective stakeholder management (Denver Airport), research-oriented development (McDonalds), poor estimation (AMR Corp.), insufficient risk management (BofA), and a host of other issues. A key objective in each postmortem should be to perform a careful analysis of what went right, what went wrong, and make recommendations that might help future project managers avoid ending up in a similar position.

The United Kingdom's National Health Service (NHS) is a prime example of an organization that has not learned from the mistakes of others. Despite the disastrous track record of other large-scale modernization projects (by the U.S. IRS, FAA, and FBI), the U.K.'s NHS elected to undertake a massive IT modernization project of its own.[6] The result is possibly the biggest and most complex technology project in the world and one that critics, including two Members of Parliament, worry may be one of the great IT disasters in the making. The project was initially budgeted at close to $12 billion. That figure is now double ($24 billion), according to the U.K. National Audit Office (NAO), the country's oversight agency. In addition, the project is two years behind schedule, giving Boston's Big Dig[7] a run for its money as the most infamous project failure of all time!

The emerging story in this case seems to be a familiar one: contractor management issues. In fact, more than a dozen vendors are working on the NHS modernization, creating a "technological Tower of Babel," and significantly hurting the bottom line of numerous companies. For example, Accenture dropped out in September 2006, handing its share of the contract to Computer Sciences Corporation, while setting aside $450 million to cover losses. Another main contractor, heath care applications maker iSoft, is on the verge of bankruptcy because of losses incurred from delays in deployment. Indeed, a great deal of time and money can be saved if we can learn from past experiences and alter our management practices going forward.

6 Initiated in 2002, the National Program for Information Technology (NPfIT) is a 10-year project to build new computer systems that are to connect more than 100,000 doctors, 380,000 nurses, and 50,000 other health-care professionals; allow for the electronic storage and retrieval of patient medical records; permit patients to set up appointments via their computers; and let doctors electronically transmit prescriptions to local pharmacies. For more information, see: http://www.baselinemag.com/article2/0,1540,2055085,00.asp and McDougall, op. cit. 2006.

7 "Big Dig" is the unofficial name of the Central Artery/Tunnel Project (CA/T). Based in the heart of Boston, MA, it is the most expensive highway project in America. Although the project was estimated at $2.8 billion in 1985, as of 2006, over $14.6 billion had been spent in federal and state tax money. The project has incurred criminal arrests, escalating costs, death, leaks, poor execution, and use of substandard materials. The Massachusetts Attorney General is demanding that contractors refund taxpayers $108 million for "shoddy work." http://wikipedia.org—retrieved on May 23, 2007.

CLASSIC MISTAKES

"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, author of *Code Complete* and *Rapid Development*

After studying the infamous failures described above, it becomes apparent that failure is seldom a result of chance. Instead, it is rooted in one, or a series of, misstep(s) by project managers. As McConnell suggests, we tend to make some mistakes more often than others. In some cases, these mistakes have a seductive appeal. Faced with a project that is behind schedule? Add more people! Want to speed up development? Cut testing! A new version of the operating system becomes available during the project? Time for an upgrade! Is one of your key contributors aggravating the rest of the team? Wait until the end of the project to fire him!

In his 1996 book, *Rapid Development*,[8] Steve McConnell enumerates three dozen classic mistakes, grouped into the four categories of people, process, product, and technology. The four categories are briefly described here:

People. Research on IT human capital issues has been steadily accumulating for over 30 years.[9] Over this time, a number of interesting findings have surfaced, including the following four:

- Undermined motivation probably has a larger effect on productivity and quality than any other factor.[10]
- After motivation, the largest influencer of productivity has probably been either the individual capabilities of the team members or the working relationships among the team members.[11]

8 McConnell, S. *Rapid Development*, Microsoft Press, 1996. Chapter 3 provides an excellent description of each mistake and category.

9 See for example: Sackman, H., Erikson, W.J., and Grant, E.E. "Exploratory Experimental Studies Comparing Online and Offline Programming Performance," *Communications of the ACM*, (11:1), January 1968, pp. 3-11; DeMarco, T., and Kister, T. *Peopleware: Productive Projects and Teams*, Dorset House, NY, 1987; Melik, R. *The Rise of the Project Workforce: Managing People and Projects in a Flat World*, Wiley, 2007.

10 See: Boehm, B. "An Experiment in Small-Scale Application Software Engineering," IEEE *Transactions on Software Engineering* (SE7: 5), September 1981, pp. 482-494.

11 See: Lakhanpal, B. "Understanding the Factors Influencing the Performance of Software Development Groups: An Exploratory Group-

- The most common complaint that team members have about their leaders is failure to take action to deal with a problem employee.[12]

- Perhaps the most classic mistake is adding people to a late project. When a project is behind, adding people can take more productivity away from the existing team members than it adds through the new ones. Fred Brooks likened adding people to a late project to pouring gasoline on a fire.[13]

Process. Process, as it applies to IT project management, includes both management processes and technical methodologies. It is actually easier to assess the effect of process on project success than to assess the effect of people on success. The Software Engineering Institute and the Project Management Institute have both done a great deal of work documenting and publicizing effective project management processes. On the flipside, common ineffective practices include:

- Wasted time in the "fuzzy front end"—the time before a project starts, the time normally spent in the approval and budgeting process.[14] It's not uncommon for a project to spend months in the fuzzy front end, due to an ineffective governance process, and then to come out of the gates with an aggressive schedule. It's much easier to save a few weeks or months in the fuzzy front end than to compress a development schedule by the same amount.

- The human tendency to underestimate and produce overly optimistic schedules sets up a project for failure by underscoping it, undermining effective planning, and shortchanging requirements determination and/or quality assurance, among other things.[15] Poor estimation also puts excessive pressure on team members, leading to lower morale and productivity.

- Insufficient risk management—that is, the failure to proactively assess and control the things that might go wrong with a project. Common risks today include lack of sponsorship, changes in stakeholder buy-in, scope creep, and contractor failure.

- Accompanying the rise in outsourcing and offshoring has been a rise in the number of cases of contractor failure.[16] Risks such as unstable requirements or ill-defined interfaces can magnify when you bring a contractor into the picture.

Product. Along with time and cost, the product dimension represents one of the fundamental trade-offs associated with virtually all projects. Product size is the largest contributor to project schedule, giving rise to such heuristics as the 80/20 rule. Following closely behind is product characteristics, where ambitious goals for performance, robustness, and reliability can soon drive a project toward failure – as in the case of the FAA's modernization effort, where the goal was 99.99999% reliability, which is referred to as "the seven nines." Common product-related mistakes include:

- Requirements gold-plating—including unnecessary product size and/or characteristics on the front end.

- Feature creep. Even if you successfully avoid requirements gold-plating, the average project experiences about a +25% change in requirements over its lifetime.[17]

- Developer gold-plating. Developers are fascinated with new technology and are sometimes anxious to try out new features, whether or not they are required in the product.

- Research-oriented development. Seymour Cray, the designer of the Cray supercomputers, once said that he does not attempt to exceed engineering limits in more than two areas at a time because the risk of failure is too high. Many IT projects could learn a lesson from Cray, including a number of the infamous failures cited above (McDonalds, Denver International Airport, and the FAA).

Technology. The final category of classic mistakes has to do with the use and misuse of modern technology. For example:

level Analysis," *Information and Software Technology* (35:8), 1993, pp. 468-474.

12 See: Larson, C., and LaFasto, F. *Teamwork: What Must Go Right, What Can Go Wrong*, Sage, Newberry Park, CA, 1989.

13 Brooks, F. *The Mythical Man-Month*, Addison-Wesley, Reading, MA, 1975.

14 Khurana, A., and Rosenthal, S.R. "Integrating the Fuzzy Front End of New Product Development," *Sloan Management Review* (38:2), Winter 1997, pp. 103-120.

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

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

17 Jones, C. *Assessment and Control of Software Risks*, Yourdon Press Series, 1994.

- Silver-bullet syndrome. When project teams latch onto a single new practice or new technology and expect it to solve their problems, they are inevitably disappointed (despite the advertised benefits). Past examples included fourth generation languages, computer-aided software engineering tools, and object-oriented development. Contemporary examples include offshoring, radio-frequency identification, and extreme programming.

- Overestimated savings from new tools or methods. Organizations seldom improve their productivity in giant leaps, no matter how many new tools or methods they adopt or how good they are. Benefits of new practices are partially offset by the learning curves associated with them, and learning to use new practices to their maximum advantage takes time. New practices also entail new risks, which people likely discover only by using them.

- Switching tools in the middle of a project. It occasionally makes sense to upgrade incrementally within the same product line, from version 3 to version 3.1 or sometimes even to version 4. But the learning curve, rework, and inevitable mistakes made with a totally new tool usually cancel out any benefits when you're in the middle of a project. (Note: this was a key issue in Bank of America's infamous failure.)

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 a research study of the lessons learned from 99 IT projects.

A META-RETROSPECTIVE OF 99 IT PROJECTS

Since summer 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. During that time, a total of 502 working professionals, each with an average of over 10 years of experience and direct involvement with at least one major IT project, have participated in the program. In partial fulfillment of program requirements, participants work in teams to conduct retrospectives of recently completed IT projects.

Thus far, a total of 99 retrospectives have been conducted in 74 different organizations. The projects studied have ranged from relatively small (several hundred thousand dollars) internally built applications to very large (multi-billion dollar) mission-critical applications involving multiple external providers. All 502 participants were instructed on how to conduct effective retrospectives and given a framework for assessing each of the following:

- Project context and description
- Project timeline
- Lessons learned—an evaluation of what went right and what went wrong during the project, including the presence of 36 "classic mistakes."
- Recommendations for the future
- Evaluation of success/failure

When viewed individually, each retrospective tells a unique story and provides a rich understanding of the project management practices used within a specific context during a specific timeframe. However, when viewed as a whole, the collection provides an opportunity to understand project management practices at a macro level (i.e., a "meta-retrospective") and generate findings that can be generalized across a wide spectrum of applications and organizations. For example, the analysis of projects completed through 2005 provided a comprehensive view of the major factors in project success.[18] That study illustrated the importance of evaluating project success from multiple dimensions, as well as from different stakeholder perspectives.

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

The first major finding was that the vast majority of the classic mistakes were categorized as either process mistakes (45%) or people mistakes (43%); see Figure 2. The remaining 12% were categorized as either product mistakes (8%) or technology mistakes (4%). None of the top 10 mistakes was a technology mistake, which confirms that technology is seldom the chief cause of project failure. Therefore, technical

18 Nelson, R. R. "Project Retrospectives: Evaluating Project Success, Failure, and Everything in Between," *MIS Quarterly Executive* (4:3), September 2005, pp. 361-372.

expertise will rarely be enough to bring a project in on-schedule, while meeting requirements. Instead, this finding suggests that project managers should be, first and foremost, experts in managing processes and people.

The second interesting finding was that scope creep didn't make the top ten mistakes. Given how often it is cited in the literature as a causal factor of project failure, this finding is surprising. Still, the fact that roughly one out of four projects experienced scope creep suggests that project managers should pay attention to it, along with its closely connected problems of requirements and developer gold-plating. Two other surprising findings were contractor failure, which was lower than expected at #13, but has been climbing in frequency in recent years, and adding people to a late project, which was #22, also lower than expected—possibly due to the impact of *The Mythical Man-Month*, by Fred Brooks.

The third interesting finding is that the top three mistakes occurred in approximately one-half of the projects examined. This finding clearly shows that if the project managers in the studied projects had focused their attention on better estimation and scheduling, stakeholder management, and risk management, they could have significantly improved the success of the majority of the projects studied.

AVOIDING CLASSIC MISTAKES THROUGH BEST PRACTICES

In addition to uncovering what went wrong on the projects studied, our retrospectives also captured what went right. We found dozens of distinct "best practices" across the 99 projects. If leveraged properly, these methods, tools, and techniques can help organizations avoid the classic mistakes from occurring in the first place. To further this intent, this section describes the top seven classic mistakes —which occurred in at least one-third of the projects —along with recommendations for avoiding each mistake.

1. Avoiding Poor Estimating and/or Scheduling

The estimation and scheduling process consists of sizing or scoping the project, estimating the effort and time required, and then developing a calendar schedule, taking into consideration such factors as resource availability, technology acquisition, and business cycles. The benefits of accurate estimates include fewer mistakes; less overtime, schedule pressure, and staff turnover; better coordination with non-development tasks; better budgeting; and, of course, more credibility for the project team. Based on the Standish Group's longitudinal findings,[19] the IT field seems to be getting somewhat better at estimating cost. In 1994, the average cost overrun was 180%. By 2003, the average had dropped to 43%. But, at the same time, the field is worse at estimating time. In 2000, average time overruns reached a low of 63%. They have since increased significantly to 82%.

Based on our research, project teams can improve estimating and scheduling by using developer-based estimates, a modified Delphi approach, historical data, algorithms (e.g., COCOMO II), and such estimation software as QSM SLIM-Estimate, SEER-SEM, and Construx Estimate.

In addition, many of the retrospective teams suggested making the estimation process a series of iterative refinements, with estimates presented in ranges that continually narrow as the project progresses over time. A graphical depiction of this concept is referred to as the estimate-convergence graph (a.k.a., the "cone of uncertainty"). A project manager creates the upper and lower bounds of the "cone" by multiplying the "most likely" single-point estimate by the optimistic factor (lower bounds) to get the optimistic estimate and by the pessimistic factor (upper bounds) to get the pessimistic estimate.

Capital One, a Fortune 500 financial services organization, uses this concept using different multipliers. Specifically, it provides a 100% cushion at the beginning of the feasibility phase, a 75% cushion in the definition phase, a 50% cushion in design, and a 25% cushion at the beginning of construction. Project managers are allowed to update their estimates at the end of each phase, which improves their accuracy throughout the development life cycle.

Four valuable approaches to improving project estimation and scheduling include (1) timebox development because shorter, smaller projects are easier to estimate, (2) creating a work breakdown structure to help size and scope projects, (3) retrospectives to capture actual size, effort and time data for use in making future project estimates, and (4) a project management office to maintain a repository of project data over time. Advocates of agile development contend that their methods facilitate better estimation and scheduling by focusing on scope,

19 Please refer to the *CHAOS Chronicles* within the Standish Group's Web site for more information: http://www.standishgroup.com.

Figure 2: Ranking of Classic Mistakes

Classic Mistakes (descending order of occurrence)	Category	No. of Projects	% of Projects
1. Poor estimation and/or scheduling	Process	51	54%
2. Ineffective stakeholder management	People	48	51%
3. Insufficient risk management	Process	45	47%
4. Insufficient planning	Process	37	39%
5. Shortchanged quality assurance	Process	35	37%
6. Weak personnel and/or team issues	People	35	37%
7. Insufficient project sponsorship	People	34	36%
8. Poor requirements determination	Process	29	31%
9. Inattention to politics	People	28	29%
10. Lack of user involvement	People	28	29%
11. Unrealistic expectations	People	26	27%
12. Undermined motivation	People	25	26%
13. Contractor failure	Process	23	24%
14. Scope creep	Product	22	23%
15. Wishful thinking	People	18	19%
16. Research-oriented development	Product	17	18%
17. Insufficient management controls	Process	16	17%
18. Friction between developers & customers	People	15	16%
19. Wasted time in the fuzzy front end	Process	14	15%
20. Code-like-hell programming	Process	13	14%
21. Heroics	People	13	14%
22. Adding people to a late project	People	9	9%
23. Silver-bullet syndrome	Technology	9	9%
24. Abandonment of planning under pressure	Process	8	8%
25. Inadequate design	Process	8	8%
26. Insufficient resources	Process	8	8%
27. Lack of automated source-code control	Technology	8	8%
28. Overestimated savings from new tools or methods	Technology	8	8%
29. Planning to catch up later	Process	8	8%
30. Requirements gold-plating	Product	8	8%
31. Push-me, pull-me negotiation	Product	5	5%
32. Switching tools in the middle of a project	Technology	5	5%
33. Developer gold-plating	Product	4	4%
34. Premature or overly frequent convergence	Process	4	4%
35. Noisy, crowded offices	People	3	3%
36. Uncontrolled problem employees	People	3	3%

short release cycles, and user stories that help estimate difficulty (rather than time).

2. Avoiding Ineffective Stakeholder Management

According to Rob Thomsett, author of *Radical Project Management*,[20] and consistent with the findings of this research, ineffective stakeholder management is the second biggest cause of project failure. For example, the challenges inherent in managing the involvement and expectations of different stakeholder groups were apparent in the following three quotes, made when a major university implemented an ERP module:

> *"I hate [the ERP application] more every day!" — Anonymous User*
>
> *"We have turned the corner ..."*
> *— Chief Operating Officer*
>
> *"In the long run, the system will be a success… Three years from now, we will be able to look back and say that we did a decent job of getting it implemented." — Project Technical Lead*

The retrospective teams identified several best practices for improving stakeholder management, including the use of a stakeholder worksheet and assessment graph.[21] This tool facilitates the three-dimensional mapping of stakeholder power (i.e., influence over others and direct control of resources), stakeholder level of interest, and stakeholder degree of support/resistance. A major project at Nextel utilized this tool to help identify the two stakeholders in most need of attention from the project management team. Failure to properly satisfy these two particular stakeholders (i.e., reduce their resistance) would have undoubtedly undermined the success of the entire project, regardless of how technologically sound the end product.

Other best practices in this area include the use of communication plans, creation of a project management office, and portfolio management. The latter two recognize the fact that some stakeholders may play roles in multiple projects. As a result, a slippage in one project may end up producing ripple effects on other projects that share one or more stakeholders.

20 Thomsett, R. *Radical Project Management*, Prentice-Hall, 2002.

21 For more information on how to construct a stakeholder worksheet and assessment graph, view the PMI webinar entitled "Political Plan in Project Management," presented by Randy Englund: http://www.pmi-issig.org/Learn/Webinars/2004/PoliticalPlaninProjectManagement/tabid/165/Default.aspx.

3. Avoiding Insufficient Risk Management

As the complexity of systems development increases, so do the number and severity of risks. The process of risk management consists of risk identification, analysis, prioritization, risk-management planning, resolution, and monitoring. Our reviews indicate that project managers rarely work their way through this process in its entirety. Thus they leave themselves in an overly reactive and vulnerable position.

Best practices uncovered in our meta-retrospective include using a prioritized risk assessment table, actively managing a top-10 risks list, and conducting interim retrospectives. Teleglobe, one of the world's leading wholesale providers of international telecommunications services, found that appointing a risk officer was useful. As with quality assurance, it is beneficial to have one person whose job is to play devil's advocate—to look for the reasons that a project might fail and keep managers and developers from ignoring risks in their planning and execution.

4. Avoiding Insufficient Planning

In the rush to develop or acquire systems that will support new business initiatives and keep top management happy, project managers too often abbreviate the planning process. An example occurred when a well-known retailer of high-end leather accessories (and a former top 10 on the *BusinessWeek* "Fastest Growing Companies" list) neglected to perform a formal business case analysis. Without an approved project charter, it proceeded with a major Web site upgrade just before the Christmas shopping season. The results:

- Clear roles and responsibilities were never established.
- Resource battles became common, negatively impacting schedule.
- Kickoff was delayed due to other projects that were wrapping up.
- Project policies, plans, and procedures were never fully developed.

The project languished until after the holiday season. Three best practices that would have helped avoid this outcome include a comprehensive project charter, clearly defined project governance, and portfolio management.

5. Avoiding Shortchanging Quality Assurance

When a project falls behind schedule, the first two areas that often get cut are testing and training. To meet deadlines, project team members often cut corners by eliminating test planning, eliminating design and code reviews, and performing only minimal testing.

In a large systems development project for the U.S. military, the review team noted the following shortcomings: the time planned for internal testing was too short; unit tests were dropped as the deadline approached; and integration testing came too late in the development cycle to complete regression testing. As a result, when the project reached its feature-complete milestone, it was too buggy to release for several more months. Shortcutting just one day of quality assurance early in a project is likely to cost 3 to 10 days downstream.[22]

Given its emphasis on testing, the retrospective teams often recommended using agile development, joint application design sessions, automated testing tools, and daily build-and-smoke tests. The latter is a process in which a software product is completely built every day and then put through a series of tests to verify its basic operations. This process produces time savings by reducing the likelihood of three common, time-consuming risks: low quality, unsuccessful integration, and poor progress visibility.

6. Avoiding Weak Personnel and/or Team Issues

Of the projects studied, 37% experienced personnel and/or team issues, which, once again, underscores the critical importance of the people dimension of project management. With regard to weak personnel, the following quote represents what occurred on several projects:

> *"One of the key problems during the development phase of the project was the relatively low skill level of the programmers assigned to the project. The weak programming skills caused schedule lapses, poor quality, and eventually caused changes in personnel."*

The cascading effects of weak personnel speak to the need to get the right people assigned to the project from the beginning and to take care of problem personnel immediately.

Another set of personnel issues seemed to stem from the trend toward outsourcing and offshoring. Between 1999 and 2006, the retrospectives reported an increasing number of problems with distributed, inter-organizational, and multi-national teams. Specific problems included a reduction in face-to-face team meetings, time-zone barriers, and language and cultural issues. In response, several teams recommended co-location as a cure, even when it required sending staff to a foreign country for an extended period of time.

7. Avoiding Insufficient Project Sponsorship

Getting top management support for a project has long been preached as a critical success factor. So it was somewhat surprising to see insufficient project sponsorship as a major issue in over one-third of the projects. On the other hand, this finding solidifies its status as a *classic* mistake. In some cases, the support was lacking from the start. In other cases, the key project sponsor departed midstream, leaving a void that was never properly filled.

The key lesson learned was the importance of identifying the *right* sponsor (typically the owner of the business process) from the very beginning, securing commitment within the project charter, and then managing the relationship throughout the life of the project (e.g., through communication plans, JAD sessions, and well-timed deliverables).

On a positive note, some projects either maintained proper sponsorship or experienced a change in sponsorship for the better. In one example of the former, the CIO responsible for implementing a mission critical, inter-organizational application made the following observation:

> *"There were CEOs on [project status] calls all the time. Nobody wanted to be written up in the 'Wall Street Journal'. That was what motivated people to change. Fear that the stock price would get hammered and fear that they would lose too much business."*

Another interesting retrospective revealed that one change in business sponsor led to the overdue cancellation of a runaway project, saving the U. S. Army, and ultimately American taxpayers, money in the long run. The review team concluded that this project was a "successful failure."

22 Jones, op. cit. 1994.

LEVERAGING IT PROJECT RETROSPECTIVES

Aggregating retrospective findings across projects and over time provides benefits, as this research study has demonstrated. Therefore we encourage managers to replicate this meta-retrospective process in their organizations. By uncovering patterns of practice, they should reap higher organizational learning, accumulate benefits over the long term, and, ultimately, increase business value.

Indeed, the collective wisdom gleaned from analyzing the 99 retrospectives suggests that mistakes tend to be people or process-related, as opposed to product or technology-related. Roughly one-half of the projects experienced problems in three areas: estimation and scheduling, stakeholder management, and risk management, while over one-third struggled with the top seven classic mistakes.

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. When instituting these best practices, it is best to cross-reference them with the classic mistakes. The matrix in Figure 3 does just that. It matches some frequently cited best practices from our retrospectives with the top 10 classic mistakes.

On the front end of a project, we encourage project managers and PMOs 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 three 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 in need of heavy user involvement should consider agile development, as a number of our retrospectives recommended.

At a more macro level, PMOs can use such a matrix to identify practices that should be incorporated into their organization's standard project methodology to avoid common mistakes across the organization.

In conclusion, the proactive and well-informed use of best practices is the best way to steer clear of classic mistakes and ultimately avoid becoming an infamous failure. For project managers, best practices are also the prescription for avoiding Einstein's definition of insanity: doing the same thing over and over and expecting different results.

Figure 3: Classic Mistakes and Best Practices Matrix

	Classic Mistakes \ Best Practices	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	Ineffective stakeholder management		X		X	X	X			X	
3	Insufficient risk 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	Insufficient project sponsorship		X		X	X	X			X	
8	Poor requirements determination	X			X						X
9	Inattention to politics		X			X	X			X	
10	Lack of user involvement	X	X		X				X	X	

ABOUT THE AUTHOR

R. Ryan Nelson

Ryan Nelson (rnelson@virginia.edu) is a Professor, the Director of the Center for the Management of Information Technology (CMIT), and the Director of the Master of Science in the Management of IT program at the McIntire School of Commerce of the University of Virginia (UVa). He received his Ph.D. in business administration from the University of Georgia in 1985 and spent five years on the faculty of the University of Houston before joining UVa. His research has been published in such journals as the *MIS Quarterly, MIS Quarterly Executive, Journal of Management Information Systems, Communications of the ACM, Information & Management, International Information Systems, Data Base,* and *CIO*. He currently serves on the editorial board of the *MIS Quarterly Executive*.

