



Looking for the Holy Grail of Software Development

Parisa Ghazi, Deimos Space

Ana M. Moreno, Universidad Politécnica de Madrid

Lawrence Peters, Software Consultants International Limited

FINDING THE KEY to successfully developing software projects has motivated the software engineering discipline from its origins in 1968. However, in software development, a concept such as “success” isn’t trivial to define. Successful software projects are traditionally described in the literature as projects that are completed within budget, on schedule, and in line with business objectives.^{1,2} But software projects are often perceived differently among the stakeholders involved.³ These perceptual differences stem from many factors, including the perceiver’s role in the project, the organizational culture, the person’s professional value system, and the system-related goals of the organization in which the stakeholders work. These and other factors result in a broad spectrum of judgments of what can be considered a successful project.

That said, a truly successful project should be one considered successful by all stakeholders. In this context, we discuss the different perceptions about software project success found in the literature. We also offer a concise snapshot of the factors these successful projects have in common to start with as well as the

commonalities they had in what they produced (we call them input and output factors, respectively). We focus on general factors related to software development instead of specific issues related to a particular method or process. Previous studies have identified a large number of issues with potential impact on projects success; we complement them with a concise view of key input and output factors.

The Literature Review

We started by performing a literature review analyzing the last decade of software engineering literature that discussed the most recurrent factors associated with successful projects.⁴ We opted for a systematic literature mapping by following the guidelines provided elsewhere.⁵ We identified two research questions:

- R.Q.1: How do different stakeholders perceive software project success?
- R.Q.2: Over the past decade, what have been the most important factors leading to successful software projects?

continued on p. 93



continued from p. 96

To answer these, we used well-known databases: ACM Portal, IEEE Xplore, and the Web of Knowledge. In parallel, we generated a list of key papers to help us check the reliability and relevancy of the searching process and the search strings. We can summarize them as $(X1 \text{ OR } X2 \dots \text{ OR } Xn) \text{ AND } (Y1 \text{ OR } Y2 \dots \text{ OR } Yn)$, where X: {successful factors, success criteria, project success} and Y: {software engineering, software projects, software development}.

We limited the search strings to the time period of 2003 to 2012 and the search itself to title, abstract, and keywords of peer-reviewed publications written in English.

From an original set of 598 papers, only 22 were finally considered as primary papers. We didn't consider those publications that were specific to a particular software method because we wanted general recommendations that developers can have in mind to increase their probability of success, as well as those that didn't provide information to either of our research questions in the title, keywords, or abstract. Nearly 90 percent of the primary studies we used were experience reports.

Success and the Eye of the Beholder

For R.Q.1, we found that the definition of a successful software project varied from one group—that is, client, software engineer, and project software manager—and culture to another among different organizations. For example, developers saw technically elegant code and design as an indication of success.⁶ Conversely, the people using the system

didn't care about elegance; they just wanted the system to do what they needed it to do in an easy-to-use, timely, reliable, and secure manner.⁷ Somewhat predictably, business-oriented stakeholders gauge success based on revenue and customers.⁸

We identified some differences among software industries. For example, in the financial services industry, good requirements and a committed sponsor were identified as critical factors, but those were among the least important factors in the consulting and telecommunications industry.⁹

Success Factors in Software Projects

For R.Q.2, after compiling the characteristics of successful software projects identified in our literature review, we identified six input factors (factors present when the project began) and four output factors (results present at the end of the project) frequently found in successful software projects. Tables 1 and 2 show these snapshots, respectively.

Table 1 confirms that manage-

ment, not technology, plays the largest role in software project success. The literature indicates that success in software projects has a great deal to do with planning, leadership, communication, teamwork, and motivation—"soft" methods that are anecdotal in nature, but crucial in practice, and not "hard" technologies, such as programming languages and methodologies. The specific technologies employed didn't enter into the prospect of success as long as they were used appropriately to achieve project goals.

Even when the choice of technology might not be trivial, software engineers are familiar with this process, and it's routinely performed during software development. But as Tom DeMarco recently claimed, "As we add new technological tools to our development process, our work becomes less, not more, technological in its focus."¹⁰ Throughout the past 40 years, although we pursued success using a bias toward technology, we often failed to vet claims and accept or reject the technology premise. At the same time, as a profession,

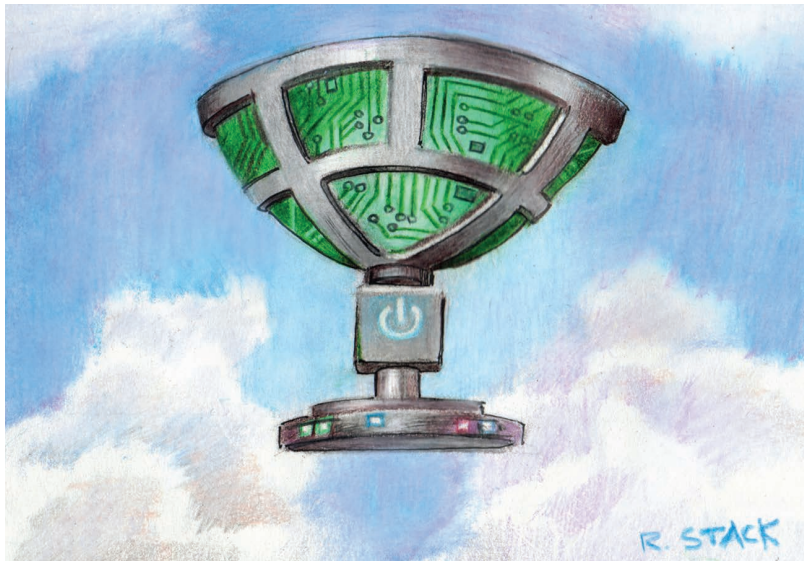


TABLE 1

Input factors of successful software projects.

Input factor	Brief description	Frequency of occurrence (%)
Clearly stated requirements	Clear and well-understood requirements accepted by the team as being realistic and achievable	50
Involved users	Active and continuous participation of users during the development process	32
Engaged, competent project manager	A project manager with the required management and leadership skills who is able to share the project's vision	23
Project planned and scheduled	A project plan and schedule developed with stakeholder participation to achieve user goals	23
Engaged, skilled team members	Competent team members with domain and technical knowledge, as well as positive attitudes about the project	18
Teamwork and communication encouraged	Development team with compatible personalities who enjoy working in a team environment and have a cooperative and mutually responsive relationship	18

TABLE 2

Output factors of successful software projects.

Output factor	Brief description	Frequency of occurrence (%)
Schedule and budget estimate maintained	Finishing the project within estimated budget and timeliness of delivery	36
Customer and user needs satisfied	Making easy-to-use, user-friendly systems that meet requirements.	23
Job satisfaction experienced on development team	Giving the development team a sense of accomplishment that sufficient quality and functionality were delivered and that they were given enough freedom and independence to be successful	18
Product quality, functionality, and performance meet high standards	Having the working product reflect the desired scope and overall quality	18

we've ignored the positive effect a competent software project manager can have on a project's success.¹¹ It was also assumed that a "cadre of competent software project managers" existed,¹² but this might not be the case.

Other relevant issues highlighted in the literature are the use of clear requirements and active participation of the user during the whole development process. Accepted and understood requirements by team members help us avoid expensive and

time-consuming rework, which helps us accomplish timely, affordable delivery of a well-tested system that meets customer/user expectations. This view of requirements is compatible with the actual idea that requirements changes, discovery, and capture go on throughout the project.¹⁰ The key is to be able to manage those changes and new features to narrow the amount of rework and the risk to cost and schedule. In this context, user participation is vital.

Regarding Table 2, as long as we

satisfy the customer's needs and do so with reliable software, we're on our way to being successful. However, one item in Table 2 might seem out of place: job satisfaction. We've known for a long time that software engineers have unique psychological profiles requiring challenging work.¹³ Without it, engagement declines, reducing motivation and productivity. If all the other output factors are present, but job satisfaction is absent, turnover can decrease, resulting in costs rising by as much as 60 per-

cent,¹⁴ schedules sliding, and the entire project being placed in jeopardy.

Here are some suggestions we've developed from our findings that can help readers more consistently achieve success:

- First and foremost, make sure that you, your development team, and your sponsors or customers agree on what constitutes success. This might take more time than you would think, but it will save time in the long run because decision making throughout the project's life cycle will go much more quickly if everyone has the same target in mind.
- While developing your project management plan, focus on all six factors listed in Table 1 and try to be objective about evaluating whether you've addressed them all. If not, even if it means delaying your project, try to do so, and if that isn't possible, be sure you manage your project in a way that mitigates their impact.
- In formulating your project plan, keep in mind how to accomplish the four output elements.

Schedule resources and activities accordingly.

- Keep in mind that the items in Tables 1 and 2 won't guarantee success but will make it more likely. Ignoring them or not addressing the potential risk involved if they aren't satisfied greatly reduces your chances of success.

If you've ever wondered what it is that successful software project managers do, addressing the issues in Tables 1 and 2 should answer your question. 🍷

References

1. C. Jones, "Patterns of Large Software Systems: Failure and Success," *Computer*, vol. 28, no. 3, 1995, pp. 86–87.
2. D. Baccharini, "The Logical Framework Method for Defining Project Success," *Project Management J.*, vol. 30, no. 4, 1999, pp. 25–32.
3. G. Klein and J. Jiang, "Seeking Consonance in Information Systems," *J. Systems & Software*, vol. 56, no. 2, 2001, pp. 195–202.
4. P. Ghazi, "What Do We Know about Successful Software Project Factors?" MS thesis, Universidad Politécnica Madrid, Spain, 2013; <http://oa.upm.es/21217>.
5. H. Thanhaim, "Team Leadership Effectiveness in Technology-Based Project Environments," *IEEE Eng. Management Rev.*, vol. 36, no. 1, 2008, pp. 165–180.
6. K. Petersen et al., "Systematic Mapping Studies in Software Engineering," *Proc. 12th Int'l Conf. Evaluation and Assessment in Software Eng.* (EASE 08), British Computer Society, 2008, pp. 68–77.
7. L.J. Peters, *Getting Results from Software Development Teams*, Microsoft Press, 2008.
8. E. Egorova et al., "Stakeholders' Perception of Success: An Empirical Investigation," *Proc. 35th Euromicro Conf. Software Eng. and Advanced Applications*, IEEE CS, 2009, pp. 210–216.
9. R. Berntsson-Svensson and A. Aybuke, "Successful Software Project and Products: An Empirical Investigation," *ACM/IEEE Int'l Symp. Empirical Software Eng.*, ACM, 2006, pp. 144–153.
10. T. DeMarco, "Bells, Whistles, Power and the Requirements Process," *IEEE Software*, vol. 30, no. 4, 2013, pp. 101–103.
11. R. Jensen, "Don't Forget about Good Management," *CrossTalk Magazine*, Aug. 2000, p. 30.
12. B. Boehm, *Software Engineering Economics*, Prentice-Hall, 1981.
13. D.J. Couger and R.A. Zawacki, *Motivating and Managing Computer Personnel*, Wiley-Interscience, 1980.
14. E. Cone, "Managing that Churning Sensation," *Information Week*, no. 680, May 1998, pp. 50–67.

PARISA GHAZI is a former software engineering MS student at Universidad Politécnica de Madrid. Contact her at parisa.ghazi@deimos-space.com.

ANA M. MORENO is full professor at Universidad Politécnica de Madrid. Contact her at ammoreno@fi.upm.es.

LAWRENCE PETERS is a software project manager and consultant. Contact him at ljpeters42@gmail.com.

IEEE Software (ISSN 0740-7459) is published bimonthly by the IEEE Computer Society. IEEE headquarters: Three Park Ave., 17th Floor, New York, NY 10016-5997. IEEE Computer Society Publications Office: 10662 Los Vaqueros Cir., Los Alamitos, CA 90720; +1 714 821 8380; fax +1 714 821 4010. IEEE Computer Society headquarters: 2001 L St., Ste. 700, Washington, DC 20036. Subscription rates: IEEE Computer Society members get the lowest rate of US\$59 per year, which includes printed issues plus online access to all issues published since 1984. Go to www.computer.org/subscribe to order and for more information on other subscription prices. Back issues: \$20 for members, \$216.17 for nonmembers (plus shipping and handling).

Postmaster: Send undelivered copies and address changes to *IEEE Software*, Membership Processing Dept., IEEE Service Center, 445 Hoes Lane, Piscataway, NJ 08854-4141. Periodicals Postage Paid at New York, NY, and at additional mailing offices. Canadian GST #125634188. Canada Post Publications Mail Agreement Number 40013885. Return undeliverable Canadian addresses to PO Box 122, Niagara Falls, ON L2E 6S8, Canada. Printed in the USA.

Reuse Rights and Reprint Permissions: Educational or personal use of this material is permitted without fee, provided such use: 1) is not made for profit; 2) includes this

notice and a full citation to the original work on the first page of the copy; and 3) does not imply IEEE endorsement of any third-party products or services. Authors and their companies are permitted to post the accepted version of IEEE-copyrighted material on their own webservers without permission, provided that the IEEE copyright notice and a full citation to the original work appear on the first screen of the posted copy. An accepted manuscript is a version which has been revised by the author to incorporate review suggestions, but not the published version with copyediting, proofreading, and formatting added by IEEE. For more information, please go to: http://www.ieee.org/publications_standards/publications/rights/paperversion-policy.html. Permission to reprint/republish this material for commercial, advertising, or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to the IEEE Intellectual Property Rights Office, 445 Hoes Lane, Piscataway, NJ 08854-4141 or pubs-permissions@ieee.org. Copyright © 2014 IEEE. All rights reserved.

Abstracting and Library Use: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use of patrons, provided the per-copy fee indicated in the code at the bottom of the first page is paid through the Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923.