

Choice of Software Development Methodologies

Do Organizational, Project, and Team Characteristics Matter?

Leo R. Vijayasarathy and Charles W. Butler,
Colorado State University

// Survey results indicate that although agile methodologies are more prevalent than 10 years ago, traditional methodologies are still popular. Organizations also use multiple methodologies on projects. Furthermore, their choice of methodologies is associated with certain organizational, project, and team characteristics. //



SOFTWARE DEVELOPMENT methodologies provide a framework for planning, executing, and managing the process of developing software systems. Many methodologies exist, including waterfall, prototyping,

iterative, rapid, structured, object-oriented, and agile methodologies. Each has its virtues, and each has its supporters and critics.

In 2003, *IEEE Software* published a special issue about the state of software engineering and software development.¹ Articles in that issue offered insights into the international use of methodologies and reflected on the practices, techniques, and tools implemented in software projects. Since then, the use of agile and hybrid methodologies has grown. It would be pertinent to discover which methodologies are common today and why organizations choose a specific methodology.

For practitioners, determining the specific methodology for a given project is critical. Sometimes, the choice of methodology might be based on marketing and literature bias that supports new or industry-supported practices. At other times, companies might rely on standards for consistency and repeatability. It's doubtful that choosing a methodology will ever be a simple deterministic exercise. Rather, the selection will likely consider several contextual factors, including organizational, project, and team characteristics, as well as market and operational forces. So, guidelines drawn from empirical associations between the methodologies used and key situational characteristics would help support informed decision making.

Toward that end, we performed a study to empirically assess the extent to which different software development methodologies, including traditional, iterative, and agile, are in use. We also sought to determine the associations between the methodologies and organizational, project, and team characteristics.

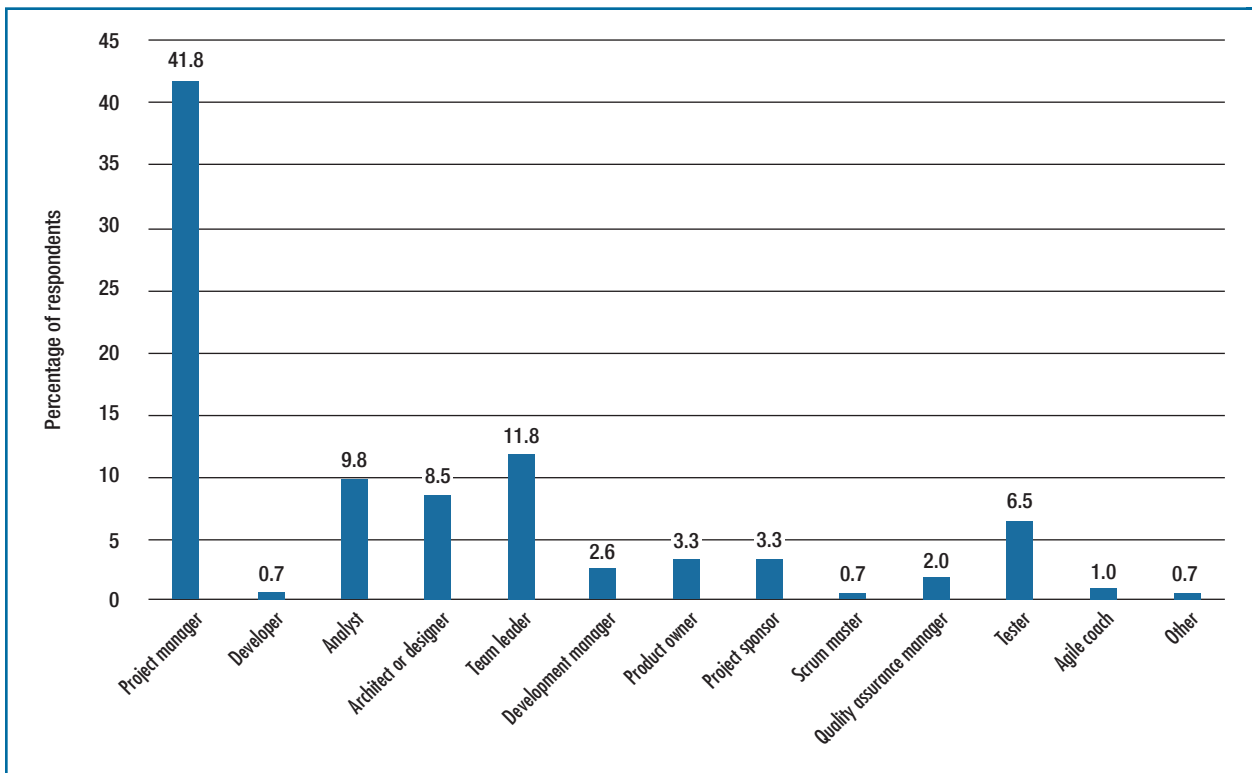


FIGURE 1. The respondents' primary project roles. Managers and leaders accounted for 58.2 percent of the roles; the technical roles (analyst, architect or designer, developer, and tester) accounted for 25.5 percent.

Online Survey and Data Profiles

We collected the study data through an anonymous online survey. Our university's Research Integrity & Compliance Review Office approved our study design and questionnaire. We posted the survey on the Project Management Institute's (www.pmi.org) website, in the Academic Research section. We also sent our survey link with a solicitation message to 2,000 project managers and team members. Our questionnaire used Qualtrics survey-building software, and it was hosted on their site. The questionnaire asked participants to base their responses on a recently completed software development project in which they had played an active role.

Respondent Profile

Figure 1 shows the respondents' primary project roles. Of the 153 respondents, 41.8 percent identified themselves as a project manager. Other common roles were team leader (11.8 percent), analyst (9.8 percent), architect or designer (8.5 percent), and tester (6.5 percent). Managers and leaders accounted for 58.2 percent of the roles; the technical roles (analyst, architect or designer, developer, and tester) accounted for 25.5 percent.

Many respondents had significant experience. Eighty-six (56.2 percent) reported more than five years' experience as a manager or lead. One hundred and two (66.7 percent) reported more than five years' experience as an analyst, architect or designer, developer, or tester.

Industry Profile

The respondents' organizations covered many industry sectors (see Figure 2). The largest percentage of respondents were from the information technology sector (26.8 percent). Other well-represented sectors were finance, banking, or insurance (13.7 percent); government or public administration (11.1 percent); professional, scientific, or technical services (8.5 percent); and the medical, dental, or healthcare professions (6.5 percent).

Respondents indicated their organizations' size in terms of the annual revenue and number of employees, using the ranges of values we provided. These measures' median ranges were US\$1 to \$100 million and 251 to 500 employees.

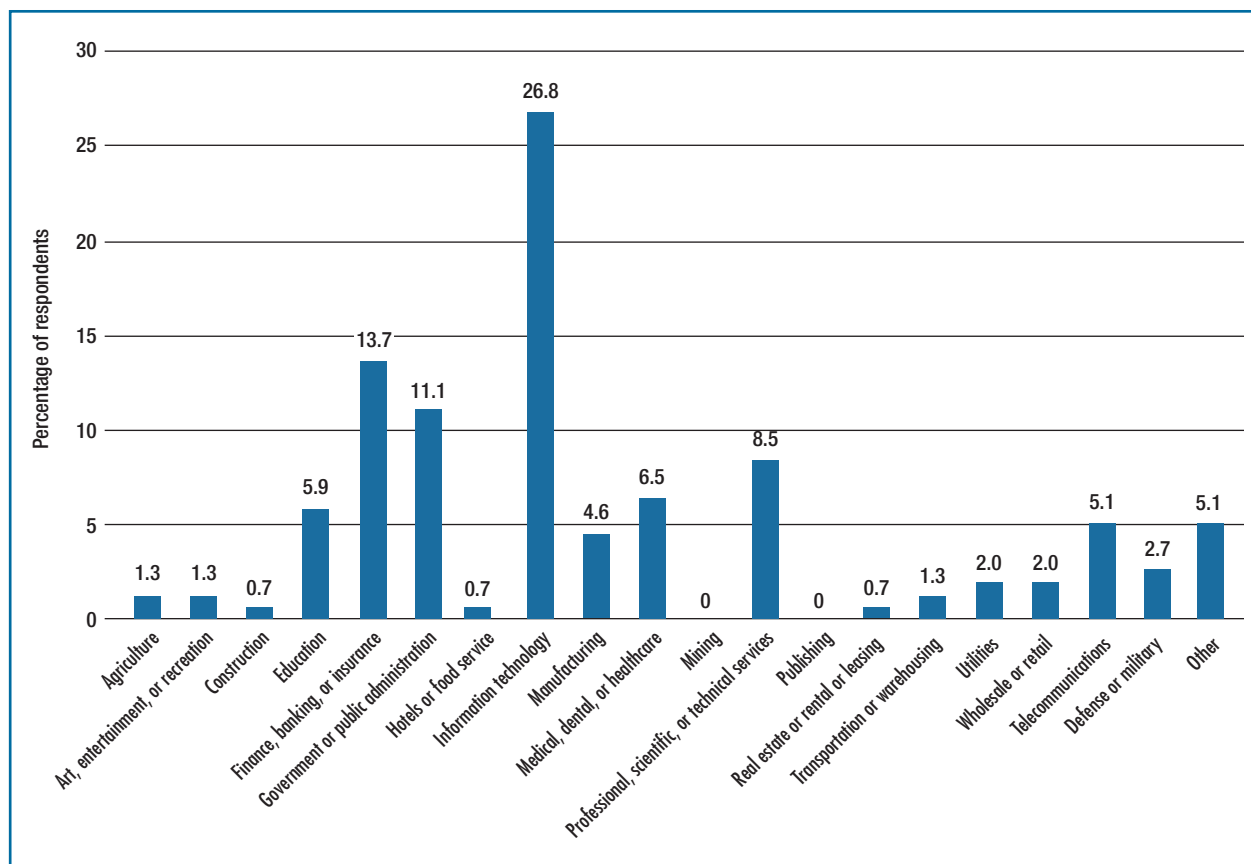


FIGURE 2. Industry distribution. About 40 percent of our respondents were from the information technology and financial, banking, or insurance sectors.

The industry sectors had a narrow geographic distribution. The US accounted for 133 respondents (86.9 percent). Other countries included India, China (including Hong Kong), the UK, Germany, Romania, Sri Lanka, France, Singapore, and Saudi Arabia.

Project Profile

The respondents worked on a variety of projects (see Figure 3a). The most common project types were new software development (38.6 percent), software enhancement (24.2 percent), customization of commercial off-the-shelf software (13.1 percent), and software integration (9.8 percent).

Figure 3b shows the project duration. Projects generally were short, with 76.4 percent lasting 12 months or less. Most of those projects took less than six months. Only 16.4 percent of the projects took 13 to 24 months, and 7.2 percent of the projects took more than 24 months. In total, 92.8 percent of the projects took 24 months or less.

Only 3.9 percent of the projects had budgets of more than \$1M. In contrast, 76.4 percent had budgets of \$400K or less. Very few budgets exceeded \$400K for project types other than new software development and software enhancement. Even so, 77 percent of new software

development project budgets were \$400K or less, and 70 percent of software enhancement project budgets were \$400K or less.

Methodologies and Approaches

Figure 4 shows the methodologies the respondents used, chosen from a list of common methodologies ranging from generic frameworks (for example, Scrum and PRINCE [*Projects in Controlled Environments*]) to more specialized processes or techniques (for example, feature-driven development and Extreme Programming). Surprisingly, the most frequently used methodology

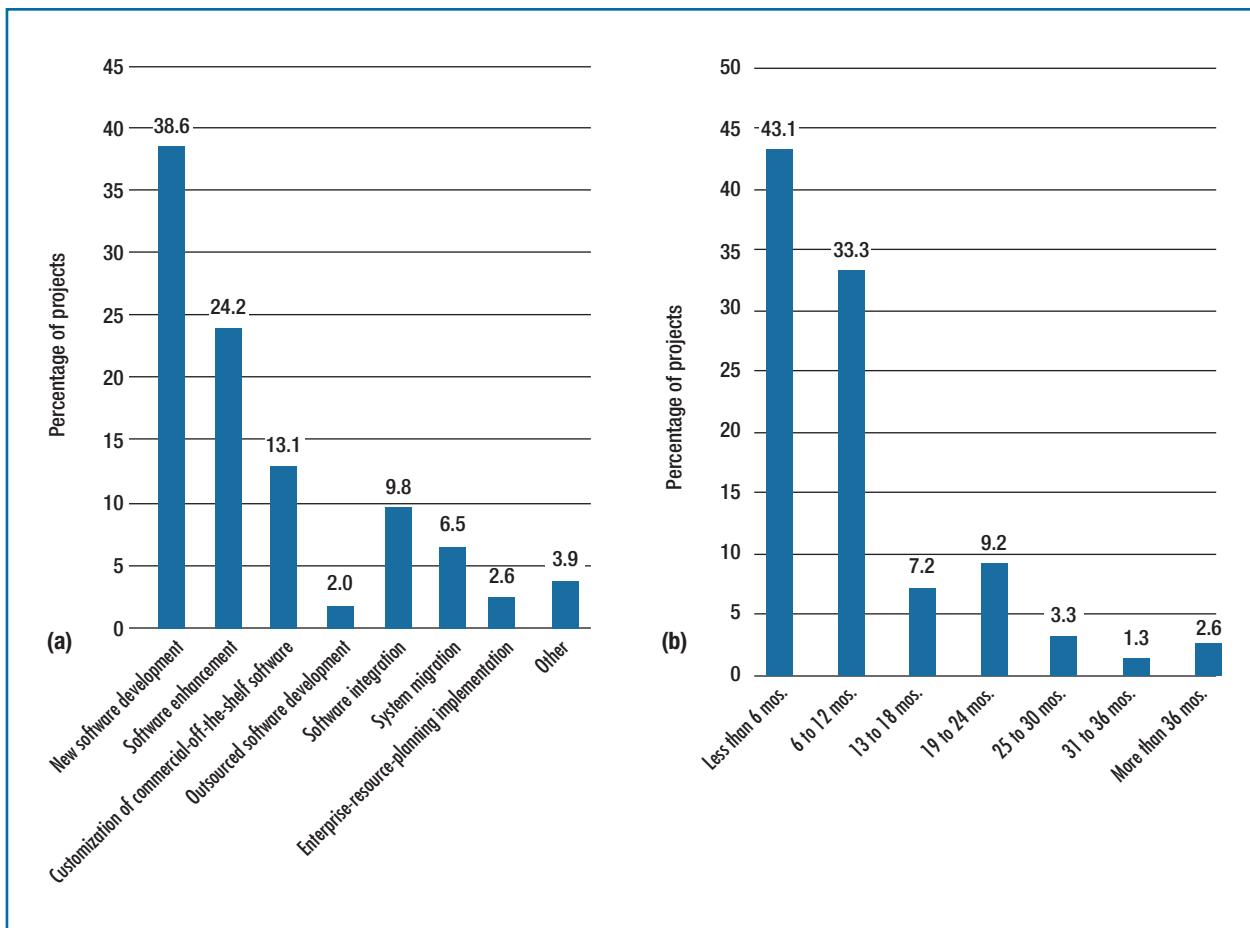


FIGURE 3. Project (a) type and (b) duration. Over half the projects developed new software or enhanced existing software; three-quarters of the projects were completed within a year.

was the waterfall model (32.0 percent). Other popular methodologies were Agile Unified Process (28.1 percent), Scrum (20.3 percent), and test-driven development (19.6 percent). None of the methodologies was used in more than one-third of the projects.

Perhaps our most interesting finding was the prevalence of a hybrid approach to methodologies; projects frequently used multiple methodologies. For example, one respondent cited using Joint Application Development for requirements gathering and the waterfall model for the

remainder of the project. So, we categorized the projects by software development approach rather than individual methodologies.

Figure 5 breaks down the projects by approach. The traditional approach segment includes projects that adopted one or more plan-driven, sequential methodologies such as the waterfall model, the structured systems analysis and design method, and PRINCE. The agile segment includes projects that used Agile Unified Process, Scrum, test-driven development, feature-driven development, adaptive software development, lean

software development, Extreme Programming, Crystal, and dynamic systems development. The iterative segment includes projects that used Rational Unified Process, Joint Application Development, and rapid application development. Finally, the hybrid segment includes projects that blended methodologies from the other segments.

These results are similar to those of prior research (see the sidebar).

Methodology Indicators

Researchers have argued that there's no silver-bullet or one-size-fits-all

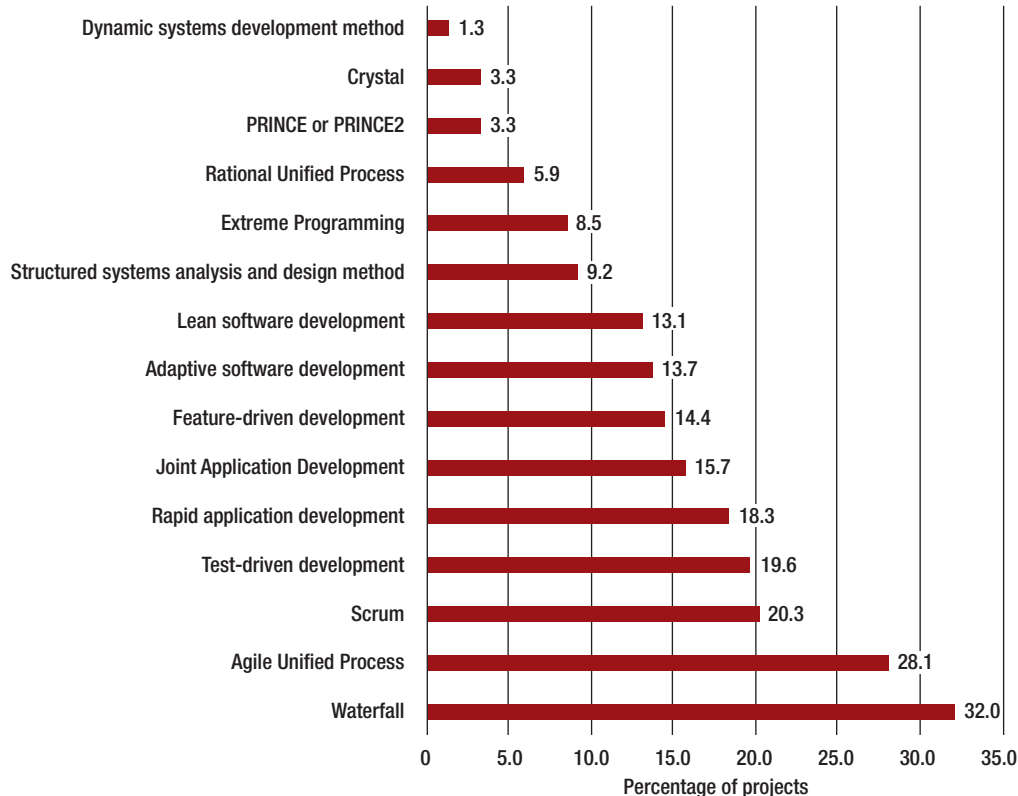


FIGURE 4. The software development methodologies used. None of the methodologies was used in more than one-third of the projects.

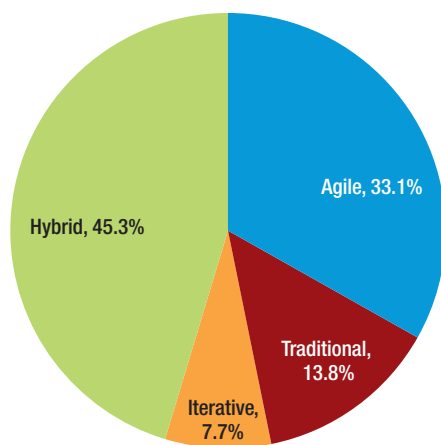


FIGURE 5. Software development approaches. Projects frequently used multiple methodologies, so we categorized the projects by software development approach rather than individual methodologies.

need to tailor processes to meet the dynamic nature of development projects⁶ and the variety of contingencies that characterize them.⁷

To determine the relationship between the development approach and organizational, project, and team factors, we used chi-square analysis with an alpha level of 0.05 for the statistical tests.

Organizational Factors

We analyzed three organizational factors: the industry, annual revenue, and number of employees. The industry wasn't statistically significant.

Figure 6a shows the distribution of companies by annual revenue for each development approach. Although large companies appeared to choose approaches differently than small companies, these differences weren't statistically significant. The most noticeable outcome was that

solution to software development.^{2–4} Paul Clark and Rory O'Connor proposed that the use of specific methods

and processes is contingent on several situational factors.⁵ In a similar vein, researchers have suggested the

RELATED WORK IN CHOICE OF DEVELOPMENT METHODOLOGIES

Because key differences exist between our study (see the main article) and prior research regarding the sampling procedure, target sample, and questionnaire items, the findings aren't directly comparable. Nevertheless, our results show a plurality in practices similar to what researchers reported more than a decade ago.

In a survey of 104 projects, Michael Cusumano and his colleagues found that many of the projects adopted both conventional practices, such as developing comprehensive specifications and detailed designs before coding, and flexible practices, such as using subcycles, daily or weekly code builds, and pair programming and testing.¹ However, unlike in our survey, even projects that had embraced "newer" techniques still widely followed traditional practices.

Colin Neill and Phillip Laplante reported that 35 percent of the 194 respondents in their study used the waterfall model.² They found that projects followed evolutionary and incremental processes even within life-cycle methodologies.

Finally, after analyzing data from approximately 12,000 software projects, Capers Jones concluded that substantial diversity in software development methods existed.³

References

1. M. Cusumano et al., "Software Development Worldwide: The State of the Practice," *IEEE Software*, vol. 20, no. 6, 2003, pp. 28–34.
2. C. J. Neill and P. A. Laplante, "Requirements Engineering: The State of the Practice," *IEEE Software*, vol. 20, no. 6, 2003, pp. 40–45.
3. C. Jones, "Variations in Software Development Practices," *IEEE Software*, vol. 20, no. 6, 2003, pp. 22–27.

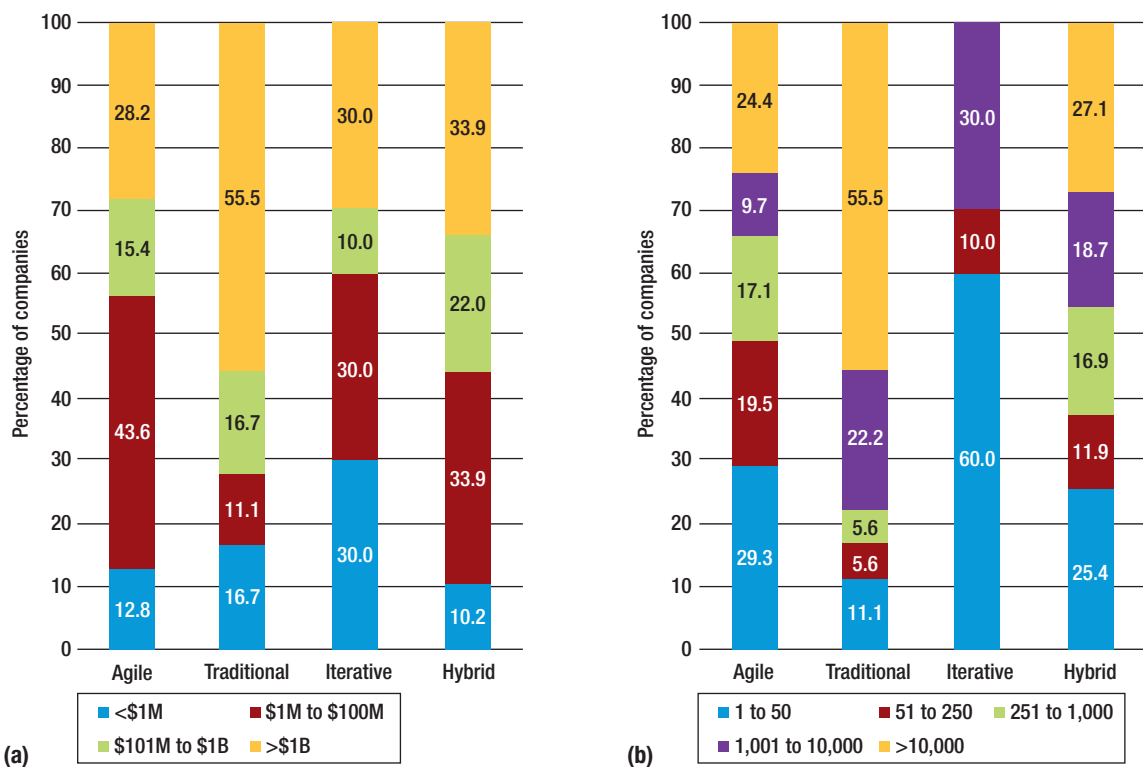


FIGURE 6. The surveyed companies according to their development approach and (a) annual revenue (US\$) and (b) number of employees. The annual revenue wasn't statistically significant; the number of employees was.

55.5 percent of the companies that used traditional approaches were high-revenue companies (>\$1B).

The number of employees was statistically significant. Figure 6b shows the distribution of companies by

employee count for each development approach. Of the companies that used traditional approaches, 55.5

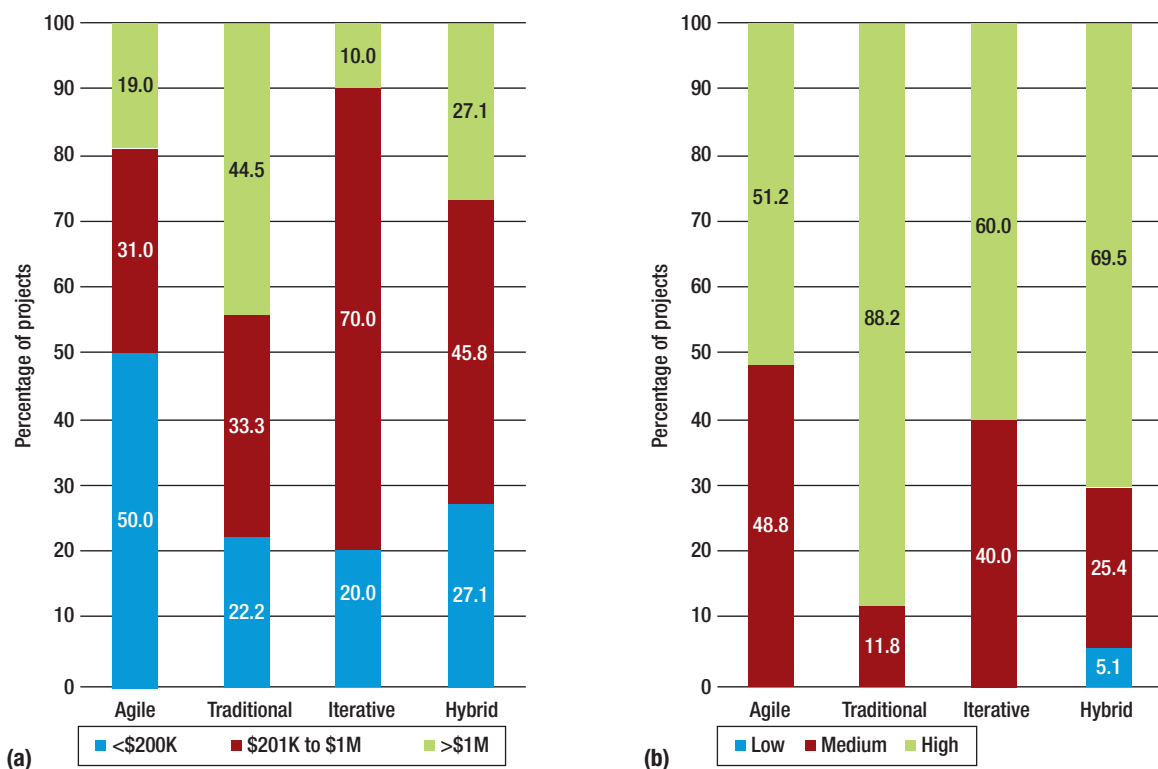


FIGURE 7. The projects according to their development approach and (a) budget and (b) criticality. Both factors had a significant association with the development approach.

percent had more than 10,000 employees, and 77.7 percent had more than 1,000 employees. In contrast, of the companies that used hybrid, agile, or iterative approaches, 45.8, 34.1, and 30.0 percent, respectively, had more than 1,000 employees. Of the companies that used iterative, agile, and hybrid approaches, 70.0, 48.8, and 37.3 percent, respectively, had 1 to 250 employees. In comparison, only 16.7 percent of companies that used traditional methodologies had 1 to 250 employees. Most companies (60 percent) that used iterative methodologies had from 1 to 50 employees.

In summary, companies with high employee counts were the dominant group for traditional approaches, whereas companies with

low employee counts were dominant for agile and iterative approaches.

Project Factors

Here, we analyzed the budget and project criticality. Both had a significant association with the development approach.

Figure 7a shows the budget-related results. Of the agile-approach projects, 50.0 percent had a budget of less than \$200K. Of the iterative- and hybrid-approach projects, 70.0 and 45.8 percent, respectively, had budgets from \$200K to \$1M. In contrast, 44.5 percent of the traditional-approach projects had budgets higher than \$1M.

We classified project criticality on the basis of responses to the survey statement, “Project success was

critical to the operations of the client’s or user’s company”:

- Low criticality reflected a response of “strongly disagree” or “moderately disagree.”
- Medium criticality reflected a response of “slightly disagree,” “neutral,” or “slightly agree.”
- High criticality reflected a response of “moderately agree” or “strongly agree.”

Figure 7b shows the results. Of all the projects, only 5.1 percent of the hybrid-approach projects had low criticality.

Medium-criticality projects accounted for 48.8, 40.0, and 25.4 percent of the projects using agile, iterative, and hybrid approaches,

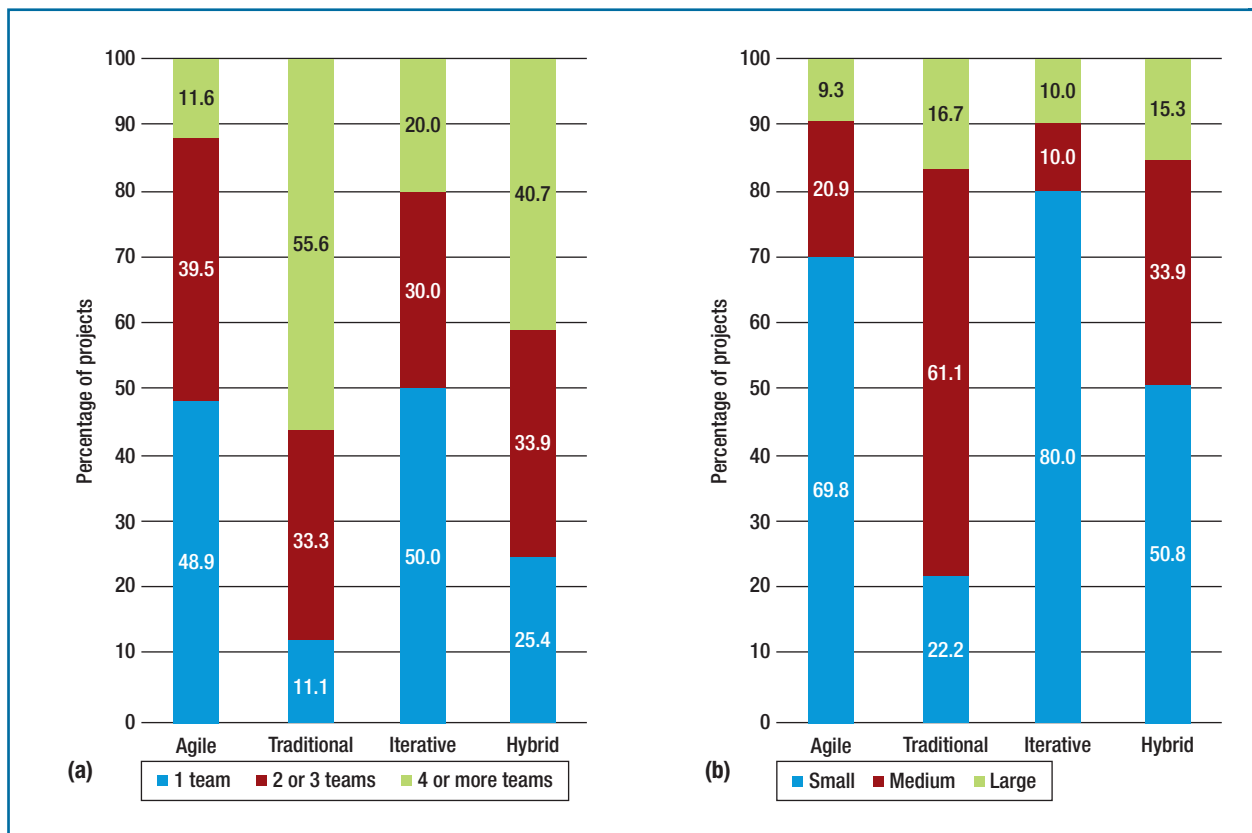


FIGURE 8. The projects according to their development approach and the (a) number of teams and (b) team size. Both factors had a significant association with the development approach.

respectively. Only 11.8 percent of the traditional-approach projects had medium criticality.

Although high-criticality projects made up the largest group for all approaches; the groups' sizes weren't uniform across the approaches. For example, 51.2 percent of agile-approach projects had high criticality, compared to 88.2 percent of the traditional-approach projects. This data suggests that organizations tend to use traditional approaches on critical projects.

Team Factors

The number of teams and team size had a significant relationship with the development approach.

Figure 8a shows the results for the number of teams. Projects with

one team were the most prevalent with agile and iterative approaches (48.9 and 50.0 percent, respectively). Projects with four or more teams were the most prevalent with traditional and hybrid approaches (55.6 and 40.7 percent, respectively). Projects with two or three teams were distributed fairly consistently, ranging from 30.0 percent for iterative approaches to 39.5 percent for agile approaches.

The results differed somewhat for team size (see Figure 8b), which we classified as small (≤ 10), medium (11 to 30), or large (> 30). Small teams were the most prevalent with agile, iterative, and hybrid approaches (69.8 percent, 80.0 percent, and 50.8 percent, respectively). Medium teams were the most prevalent


with traditional approaches (61.1 percent). Large teams were rare, ranging from 9.3 percent for agile approaches to 16.7 percent for traditional approaches.

Our survey has two main takeaways. First, although no particular methodology was used in more than one-third of the projects, agile methodologies had high use—measurably higher than in 2003, when less than 5 percent of surveyed developers cited an agile approach.³ Also, Scrum, Agile Unified Process, and test-driven development were three of the top four most frequently cited methodologies. In addition, hybrid approaches frequently used agile methodologies.

TABLE 1

Characteristics of projects following the four software development approaches.

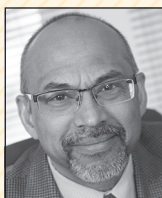
Approach	Characteristics		
	Organizational	Project	Team
Agile	Moderate revenue A small number of employees	Low budget Medium to high criticality	One team Small team
Traditional	High revenue A large number of employees	High budget High criticality	Multiple teams Medium team
Iterative	A small number of employees	Medium budget Medium to high criticality	One team Small team
Hybrid	Organization size unimportant	Medium budget High criticality	Small team

methodologies to ensure positive project outcomes. The knowledge gained from such research can help identify the best methodologies for different contexts. 

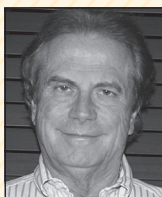
References

1. R.L. Glass, "The State of the Practice of Software Engineering," *IEEE Software*, vol. 20, no. 6, 2003, pp. 20–21.
2. F.P. Brooks, "No Silver Bullet—Essence and Accident in Software Engineering," *Proc. IFIP 10th World Computing Conf.*, 1986, pp. 1069–1076.
3. O. Benediktsson, D. Dalcher, and H. Thorbergsson, "Comparison of Software Development Life Cycles: A Multiproject Experiment," *IEEE Proc. Software*, vol. 153, no. 3, 2006, pp. 87–101.
4. B. Boehm and R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed*, Pearson Education, 2003.
5. P. Clarke and R.V. O'Connor, "The Situational Factors That Affect the Software Development Process: Towards a Comprehensive Reference Framework," *Information and Software Technology*, vol. 54, no. 5, 2012, pp. 433–447.
6. P. Xu and B. Ramesh, "Using Process Tailoring to Manage Software Development Challenges," *IT Professional*, vol. 10, no. 4, 2008, pp. 39–45.
7. G. Kalus and M. Kuhrmann, "Criteria for Software Process Tailoring: A Systematic Review," *Proc. 2013 Int'l Conf. Software and System Process*, 2013, pp. 171–180.

ABOUT THE AUTHORS



LEO R. VIJAYASARATHY is an associate professor of computer information systems in Colorado State University's College of Business. His research explores the development, use, and consequences of information systems. Vijayasaraty received a PhD in management information systems from Florida International University. Contact him at leo.vijayasaraty@colostate.edu.



CHARLES W. BUTLER is a professor of computer information systems in Colorado State University's College of Business. His research explores software development, software metrics, and project management. Butler received a PhD in business from Texas A&M University. Contact him at charles.butler@colostate.edu.

Second, select organizational, project, and team characteristics had strong associations with specific approaches (see Table 1). These associations offer insights into how some key situational factors might influence the choice of software development methodologies.

However, more empirical research is necessary. Researchers need to examine the role of other factors such as the degree of business and technology dynamism, uncertainty, and risk. They also need to ascertain the optimal fit between contingent factors and



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.