Choice of Software Development Methodologies – Do Project, Team and Organizational Characteristics Matter?

Leo R. Vijayasarathy and Charles W. Butler

Colorado State University

Abstract

Organizations have a number of methodological options for software development ranging from traditional to agile approaches. We conducted a survey of project managers and other team members about their use of software development methodologies. Our results indicate that while agile approaches such as agile unified process and scrum are more prevalent than 10 years ago, traditional methodologies including waterfall are still popular. Interestingly, organizations are also adopting multiple methodologies on projects and choosing to follow a hybrid approach to software development. Further, their choice of methodologies is associated with certain project, team and organizational characteristics.

Key Words: software development methodology, agile methodology, traditional methodology, hybrid software development approaches, project characteristics, team characteristics, organizational characteristics

Author Biographies

Leo R. Vijayasarathy is Associate Professor of Computer Information Systems in the College of Business at Colorado State University. His research explores the development, use and consequences of information systems. He earned an MBA from Marquette University and Ph.D. from Florida International University. Contact him at 155 Rockwell Hall, Colorado State University, Fort Collins, CO 80523; 970-491-0607; leo.vijayasarathy@colostate.edu.

Charles W. Butler is Professor of Computer Information Systems in the College of Business at Colorado State University. His research explores the software development, software metrics and project management. He earned an MS from the University of South Florida and Ph.D. from Texas A&M University. Contact him at 019 Rockwell Hall, Colorado State University, Fort Collins, CO 80523; 970-491-6436; charles.butler@colostate.edu.

Digital Object Indentifier 10.1109/MS.2015.26 0740-7459/$26.00 2015 IEEE

Introduction

Software development methodologies provide a framework for planning, executing, and managing the process of developing software systems. There are many software development methodologies including waterfall, prototyping, iterative, rapid, structured, object, and agile approaches. Each of these approaches has its virtues, and each also 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 this 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, and it would be pertinent to discover which software development methodologies are used more frequently today and why a specific methodology is chosen.

For practitioners, determining the specific software development methodology to use for a given project is a critical decision. Sometimes, the decision to use a software development methodology maybe based on marketing and literature bias which support new or industry supported practices. At other times, companies may rely on standards for consistency and repeatability. It is doubtful that the choice of a software development methodology will ever be a simple deterministic exercise. Rather, a number of contextual factors including organizational, project, and team characteristics as well as market and operational forces are likely to be considered in the selection process. Therefore, guidelines drawn from empirical associations between methodologies used and key situational characteristics would be valuable for informed decision making.

Objectives of Current Study

Our study has two objectives. First, we wanted to empirically assess the extent to which different software development methodologies including traditional, iterative, and agile are in use today. Second, we sought to determine if there are discernible associations between certain organizational, project, and team characteristics and the methodologies used.

Survey Design and Data Collection

Data for this study was collected through an anonymous online survey employing both a pull and push strategy. Our study design and questionnaire were approved by our university's Research Integrity and Compliance Review office. A web-based survey was posted on the Project Management Institute's (www.pmi.org) website within the Academic Research section. We also sent our survey link with a solicitation message to 2,000 software development project managers and team members including analysts, designers, developers and testers. Our questionnaire used Qualtrics survey building software and it was hosted at their site. The questionnaire asked participants to base their responses with reference to a recently completed software development project in which they had played an active role.

Respondent Profile

As shown in Figure 1, a total of 153 responses were completed from software development project team members, 41.8% of whom identified their role as project manager in the project they chose to base their survey responses. Other frequently cited project roles were team leader

Digital Object Indentifier 10.1109/MS.2015.26 0740-7459/$26.00 2015 IEEE

(11.8%), analyst (9.8%), architect/designer (8.5%), and tester (6.5%). Generally, managers and leaders accounted for 58.2% of the project roles, and the technical project roles including analysts, designers, developers, and testers accounted for 25.5% of the roles. Many of the respondents had significant experience as managers and leaders. Eighty-six respondents (56.2%) reported more than 5 years of experience as a manager or lead. One hundred and two respondents (66.7%) reported more than 5 years of experience as analysts, designers, developers or testers.

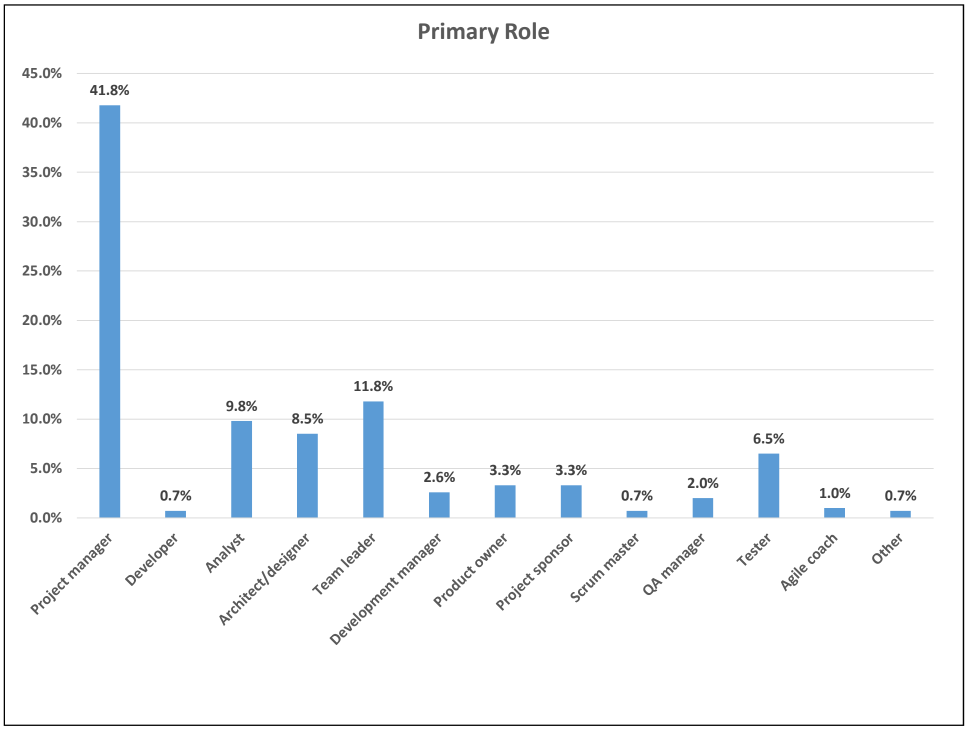

Figure 1 Respondent Primary Role

Industry Profile

Respondent organizations cover many different industry sectors. As shown in Figure 2, the largest percentage of projects was from companies in the Information Technology sector (26.8%). Finance/Banking/Insurance (13.7%), Government/Public Administration (11.1%), Professional, Scientific and Technical Services (8.5%), and Medical/Dental/Healthcare (6.5%) were well-represented industries. Respondents indicated the size of their organizations in terms of annual revenues and number of employees from ranges of values provided for both measures. The median ranges for these measures were US$ 1 to 100 million and 251 to 500 employees. These industry sectors were located in a narrow geographic distribution. One hundred and thirty-three respondents, 86.9% of all respondents, cited the United States as their country or geographic area. Other countries stated as respondent residency included India, China, United Kingdom, Germany, Romania, Sri Lanka, France, Singapore, Hong Kong and Saudi Arabia.

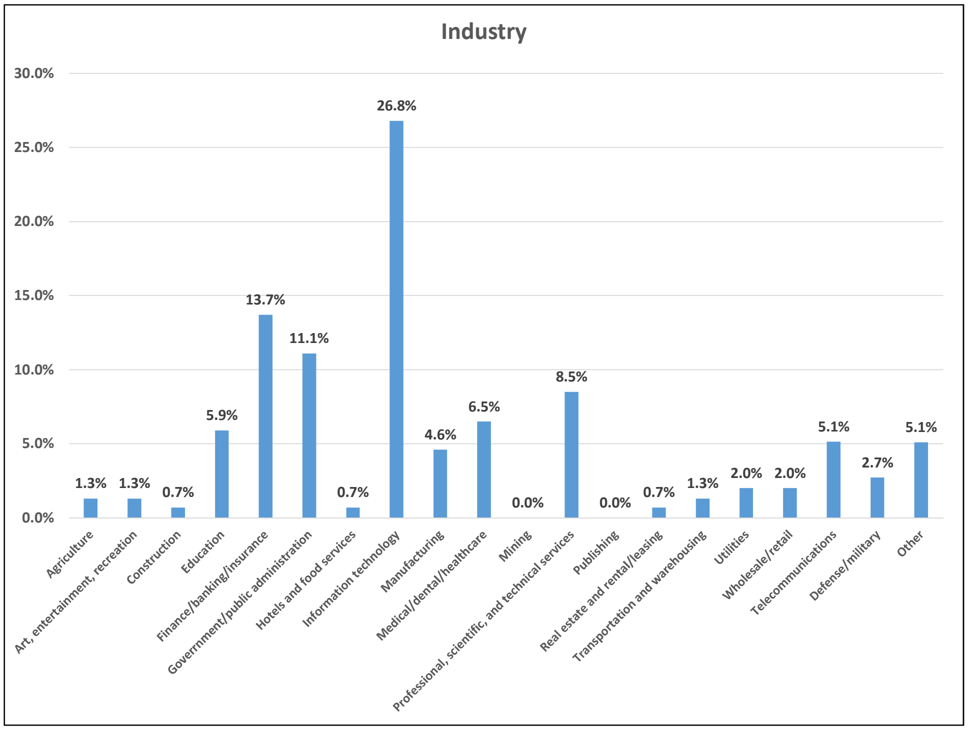

Figure 2 Industry Distribution

Project Profile

The respondents worked on a variety of project types. The most frequent project type selected was new software development, accounting for 38.6% of the total number of projects (see Figure 3). Software enhancement was chosen by 24.2% of the respondents indicating that over half of the projects were related to software development. Customization of commercial off-the-shelf software was also a frequent project type, cited by 13.1% of the respondents, followed closely by software integration which accounted for 9.8% of the projects.

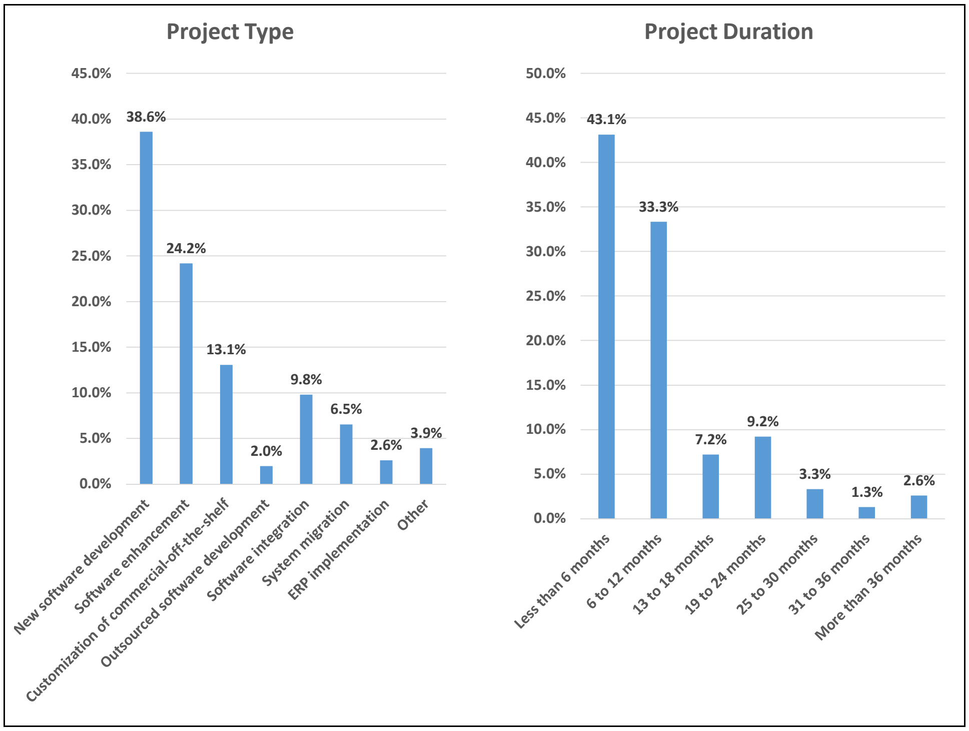

Figure 3 Project Attributes

As displayed in Figure 3, project durations were generally short with 76.4% of respondents stating their project duration were less than 12 months. Over one half of these projects were completed in less than six months. For projects with duration of more than 12 months, the number of longer term projects fell off measurably. Over 16.0% of the projects were completed within one to two years, and 7.2% of the projects took longer than two years to complete. In total, 92.8% of all project durations were 24 months or less, indicating long duration projects were not popular among the respondents.

Along with relatively short project durations, budgets provide insight into project execution. Only 3.9% of the projects had budgets over $1M. In contrast, 76.4% of the projects had budgets $400K or less. An interesting observation about project types and budgets is that there were very few budgets over $400K for project types other than new software development and software enhancement. Even so, 77% of new software development project budgets were $400K or less, and 70% of software enhancement budgets were $400K or less, denoting low budgets for respondent projects.

Software Development Methodologies

As Figure 4 indicates, respondents identified the methodologies that were used in their software development projects from a list of methods that is commonly associated with developing software, ranging from generic frameworks (e.g., Scrum, Prince) to more specialized processes/techniques (e.g., Feature-driven development, eXtreme Programming). Surprisingly, the systems development lifecycle (waterfall) was the most frequently used methodology cited by 32.0% of the respondents. Other popular methodologies were agile unified process (AUP), scrum and test-driven development (TDD), which were used in 28.1%, 20.3%, and 19.6% of the projects, respectively. No methodology was dominant and waterfall methodology still maintains an active role in industry.

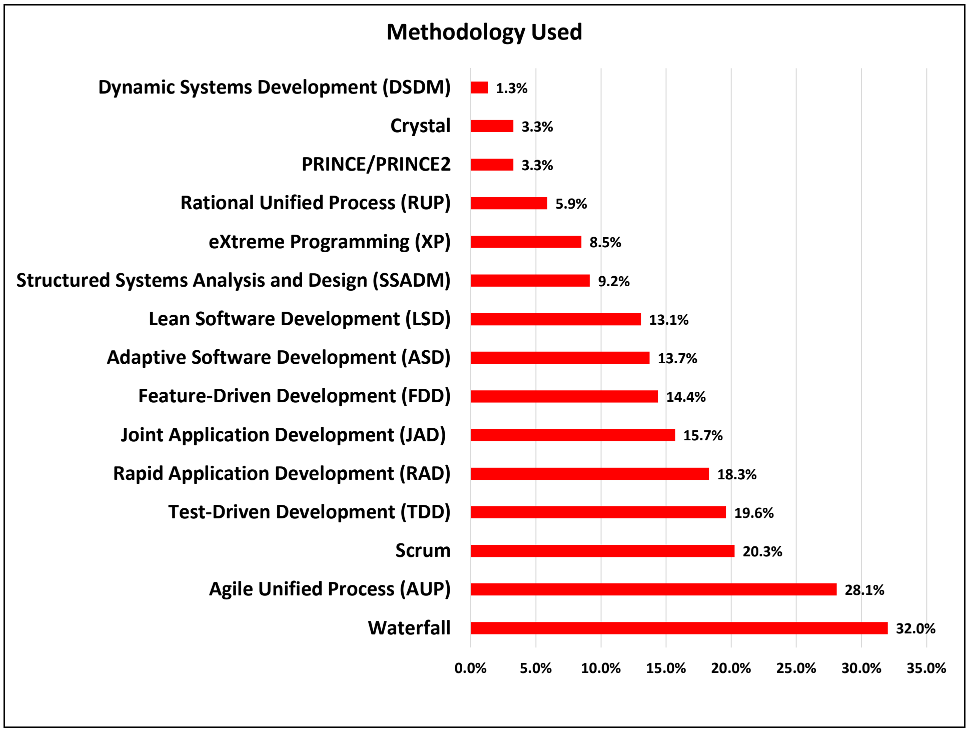

Figure 4 Cited Usage of Software Development Methodologies

Approach to Software Development

Perhaps the most interesting finding we uncovered in this study was the prevalence of a hybrid approach in the software development methodologies used on projects. Since the respondents were asked to identify one or more methodologies that were used in their projects, we discovered that projects frequently used multiple software development methodologies. For example, one respondent cited the use of JAD for requirements gathering and waterfall for the remainder of the project. This finding led us categorize the projects by software development approach rather than individual software development methodologies.

Digital Object Indentifier 10.1109/MS.2015.26 0740-7459/$26.00 2015 IEEE

A breakdown of the projects by software development approach is presented in Figure 5. The traditional approach cluster represents projects that adopted methodologies that are plan-driven and sequential in nature. These projects used one or more of the following methodologies: waterfall, SSADM and PRINCE. Projects that followed methodologies embodying the agile manifesto and principles (agilemanifesto.org) are grouped under the agile cluster. The methodologies included in this group are AUP, scrum, TDD, FDD, ASD, LSD, XP, crystal and DSDM. Projects that employed methodologies with an emphasis on iterative development are classified as belonging to the iterative cluster, and include projects that used RUP, JAD and RAD. Finally, projects that had blended methodologies from two or more of the first three clusters (i.e., traditional, agile and iterative), make up the hybrid cluster.

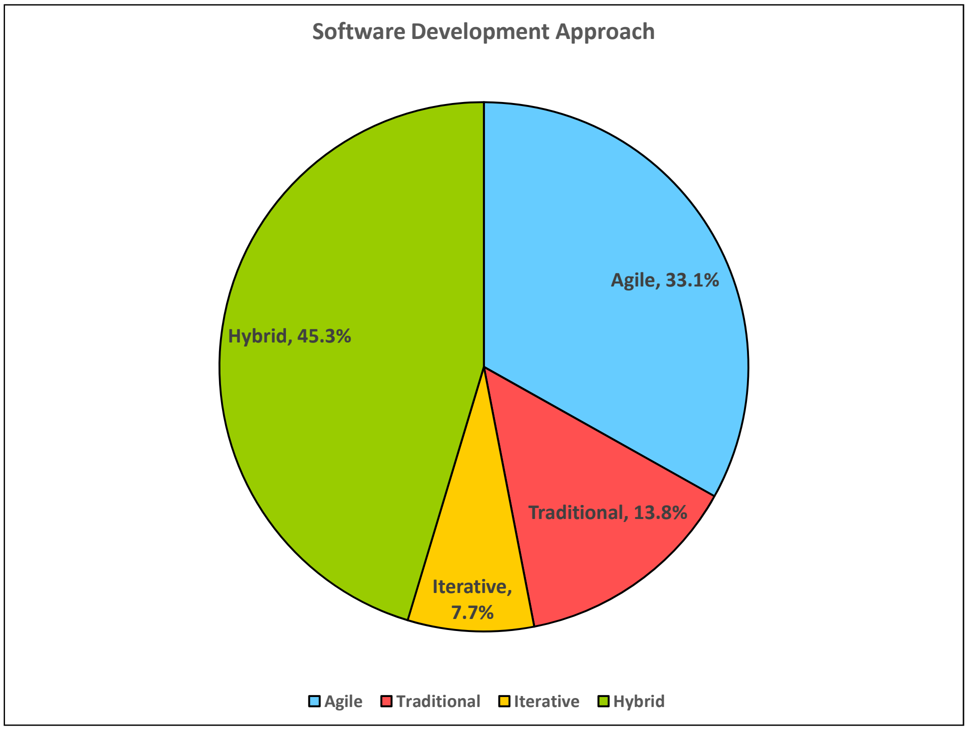

Figure 5. Software Development Approach Clusters

Comparison to Prior Studies

Since there are key differences in the sampling procedure, target sample and questionnaire items between our study and prior research, the findings cannot be directly compared. Nevertheless, our results show a similar pattern of plurality in practices that were reported more than a decade ago.[2,3,4] From their survey of 104 projects, Cusumano and his colleagues found that many projects in their sample adopted both conventional practices such as the development of comprehensive specifications and detailed designs before coding as well as flexible practices such as the use of subcycles, daily or weekly code builds and pair programming and testing. However, in contrast to our results, traditional practices were more widely followed even on projects that had embraced "newer" techniques. Similar results were reported by Neill and Laplante, who found that the waterfall methodology was used by 35% of its 194 respondents, and that evolutionary and incremental processes were followed even within lifecycle methodologies.

Again, allowing for differences between our studies, our findings also echo the observations of Jones, who analyzed data from about 12,000 software projects and concluded that there was substantial diversity in software development methods.

Software Development Methodology Indicators

Researchers[5, 6, 7] have argued that there is no "silver bullet" or "one size fits all" solution to software development, and have proposed that the use of specific methods and processes are contingent on a number of situational factors.[8] In a similar vein, authors have suggested the need to tailor processes to meet the dynamic nature of software development projects[9] and the wide variety of contingencies that characterize them[10]. We investigated whether there are significant relationships between the software development approach and relevant contextual factors related to the organization, project and team, using chi-square analysis with an alpha level of 0.05 for the statistical tests.

Organizational Factors

Three organizational factors, industry, annual revenues and number of employees, were analyzed for possible associations with the software development approach used. Industry was not statistically significant. The left bar chart in Figure 6 shows the distribution of companies by annual revenues for each software development approach. Although large companies appear to operate differently than small companies in their choice of software development approaches, these differences were not statistically significant. The most noticeable descriptive outcome for annual revenues was that 55.6% of the companies that used traditional methodologies were high revenue companies (revenues > US$1B).

While annual revenues was not statistically significant, the number of employees was. The right bar chart in Figure 6 shows the distribution of companies by employee count for each software development approach. Among companies that used traditional approaches for their projects, 55.6% had more than 10,000 employees, and about 77.8% of the companies adopting traditional methodologies had more than 1,000 employees. In contrast, among those companies that used hybrid, agile and iterative approaches, 45.7%, 34.2% and 30%, respectively had more than 1000 employees. Most of the companies (60%) that used iterative methodologies had between 1 to 50 employees. Overall, companies with 1 to 250 employees account for 48.8%, 70% and 37.3% of the companies that used agile, iterative and hybrid approaches, respectively. In comparison, only 16.7% of companies of similar size had used traditional methodologies for their projects. In summary, companies with high employee counts were the dominant group for the traditional approach while companies with low employee counts were most prominent for agile and iterative approaches.

Digital Object Indentifier 10.1109/MS.2015.26 0740-7459/$26.00 2015 IEEE

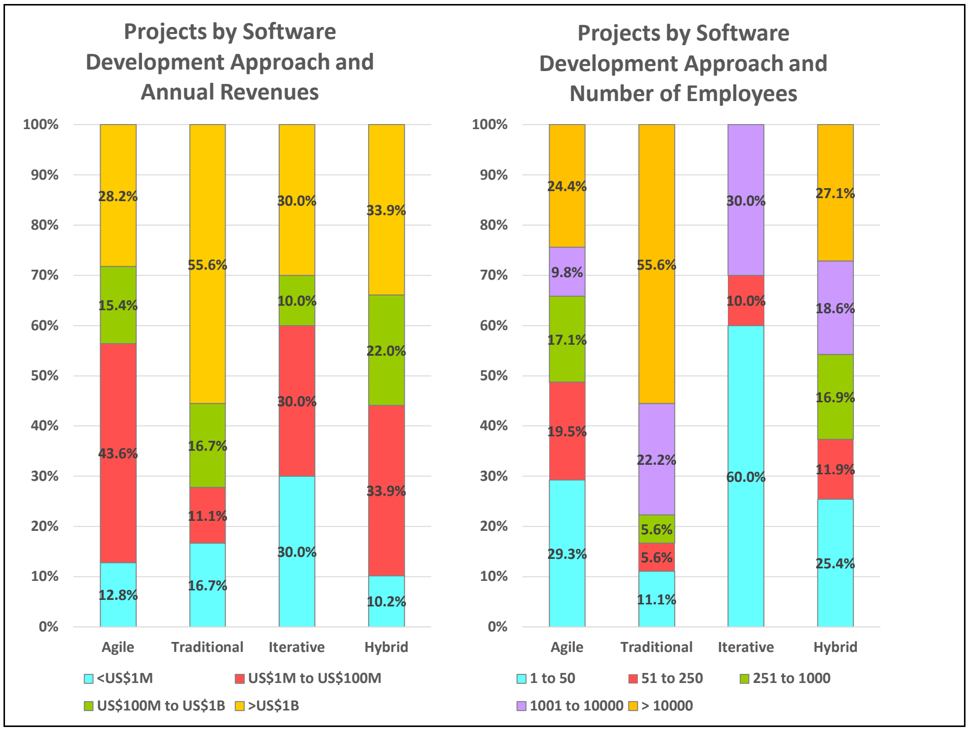

Figure 6. Organizational Factors

Project Factors

We examined two project-related factors variables, budget and project criticality. Both factors had significant associations with the development approaches. There are observable partialities for the software development approach used based upon budget magnitude (Figure 7). Among projects that used an agile approach, 50% had a budget of less than US$250K. For projects that followed iterative methodologies, a strong majority (70%) had budgets between US$200K and US$1M. Similarly, among projects that adopted a hybrid approach, a sizable percentage (45.8%) also had mid-sized budgets ranging from US$200K and US$1M. In contrast, projects that favored traditional methodologies, 44.4% had budgets in excess of US$1M.

We classified project criticality from responses on a seven-point Likert scale to the survey question, “Project success was critical to the operations of the client/user company” (low – strongly and moderately disagree; medium – slightly disagree, neutral or slightly agree; high – moderately or strongly agree). As shown in Figure 7, there are distinct patterns in project criticality depending on the development approach used. First, there were no projects classified as having low criticality except among projects that used hybrid methodologies. Second, for agile, iterative and hybrid approaches, projects that had medium to low criticality accounted for 48.8%, 40.0% and 30.5%, respectively. In contrast, only 11.8% of the projects that used a traditional approach were considered to be of medium criticality. Third, although projects that had high criticality made up the largest group for all approaches; the sizes of the groups were not uniform across the approaches. The largest contrast was between agile and traditional approaches with 51.2% of projects with high criticality adopting the former and 88.2% of projects with the same criticality following the latter. From this data, it looks as if organizations tend to use a traditional software development when a project is critical.

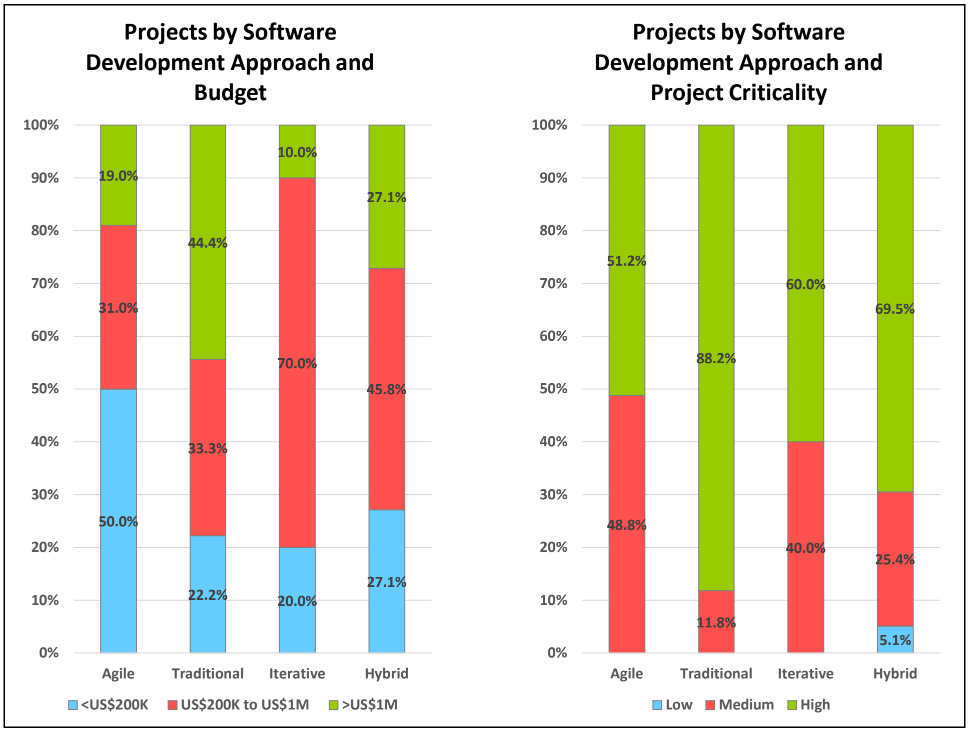

Figure 7 Project Factors

Team Factors

Figure 8 shows two team factors, number of teams and team size, that had significant relationships with software development approaches. Among projects that used agile and iterative methodologies, the majority of them (48.8% and 50.0%, respectively) was composed of a single team. In contrast, projects with 4 or more teams were more prominent within the traditional (55.6%) and hybrid (40.7%) approaches. Projects with 2 or 3 teams were indifferent regarding the choice of software development approach, since teams of this size accounted for approximately the same percentage (ranging from 30% for iterative to 39.5% for agile) within each approach. The results are somewhat different for team size (small - ≤ 10 members; medium – 11 to 30 members and large - > 30 members). For agile, iterative, and hybrid approaches, a strong majority of projects deployed small teams, 69.8%, 80.0%, and 50.8%, respectively. In comparison, medium sized teams are the most prevalent (61.1%) in projects that utilized traditional development approaches. Interestingly, large teams were rare, regardless of the software development approach - 9.3%, 16.7%, 10.0%, and 15.3% for agile, traditional, iterative, and hybrid approaches, respectively.

Digital Object Indentifier 10.1109/MS.2015.26 0740-7459/$26.00 2015 IEEE

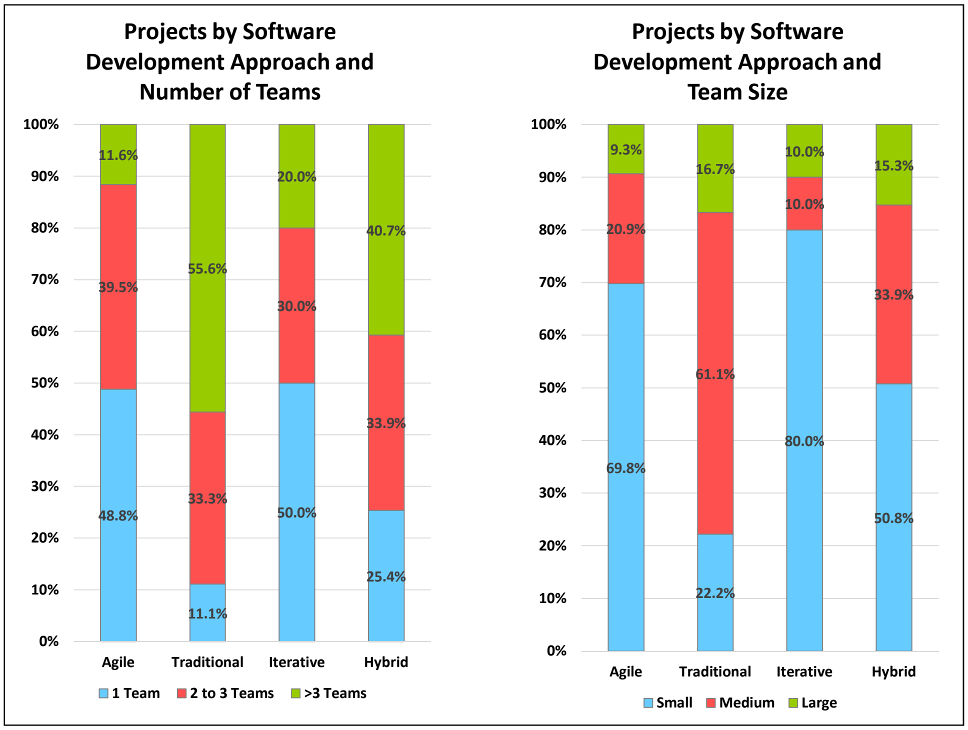

Figure 8 Team Factors

Summary

There are important takeaways from our survey. First, with regard to the state of software development methodology practice, there is not one particular approach that dominates. However, agile methodologies as a group have an observably high level of usage, and their usage is measurably higher than in 2003 when less than 5% cited an agile approach.[3] Scrum, agile unified process, and test-driven development are three of the top four most frequently citied methodologies. In addition, agile methodologies are frequently used in hybrid approaches.

Second, there are select organizational, project and team attributes that have strong associations with the specific methodologies used. In brief, projects executed with each of the four development approaches tend to exhibit the following organizational, project and team characteristics:

- **Agile:** organizations with moderate revenues and small number of employees; projects with low budgets and medium to high project criticality; one team and small team size.
- **Traditional:** organizations with high revenues and large number of employees; projects with high budgets and high project criticality; multiple teams and medium team size.
- **Iterative:** organizations with small number of employees; projects with medium budgets and medium to high criticality; one team and small team size.
- **Hybrid:** organizational size does not matter, projects with medium budgets and high project criticality; small team size.

Digital Object Indentifier 10.1109/MS.2015.26 0740-7459/$26.00 2015 IEEE

These associations offer insights into the possible influence of some key situational factors in the choice of software development methodologies. However, more empirical work is needed; not only to examine the role of other factors such as the degree of business and technology dynamism, uncertainty and risk, but also to ascertain the optimal fit between contingent factors and methodologies to ensure positive project outcomes. The knowledge gained from this research stream can help identify the best frameworks for planning, executing, and managing the development of software systems under different contextual settings.

References

1. R.L. Glass, "The State of the Practice of Software Engineering," *IEEE Software*, vol. 20, no. 6, 2003, pp. 20-21.

2. M. Cusumano, A. MacCormack, C. F. Kemerer and B. Crandall, "Software Development Worldwide: The State of the Practice," *IEEE Software,* vol. 20 , no. 6, 2003, pp. 28-34.

3. C. J. Neill and P. A. Laplante, "Requirements Engineering: The State of the Practice," *IEEE Software,* vol. 20, no. 6, 2003, pp. 40-45.

4. C. Jones, "Variations in Software Development Practices," *IEEE Software,* vol. 20, no. 6, 2003, pp. 22.27.

5. F.P. Brooks, "No Silver Bullet – Essence and Accicent in Software Engineering," *Proceedings of the IFIP Tenth World Computing Conference*, H.-J. Kugler, ed., Elsevier Scince B.V., Amsterdam, NL, 1986, pp. 1069-76.

6. O. Benediktsson, D. Dalcher and H. Thorbergsson, "Comparison of Software Development Life Cycles: A Multiproject Experiment," *IEE Proceedings – Software*, vol. 153, no. 3, 2006, pp. 87-101.

7. B. Bohem and R. Turner, *Balancing Agility and Discipline – A Guide for the Perplexed*, Pearson Education, Boston, MA, USA, 2003.

8. 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.

9. P. Xu and B. Ramesh, "Using Process Tailoring to Manage Software Development Challenges," *IT Professional*, vol. 10, no. 4, 2008, pp. 39-45.

10. G. Kalus and M. Kuhrmann, "Criteria for Software Process Tailoring: A Systematic Review," *Proceedings of the International Conference on Software and System Process*, ACM, New York, NY, USA, 2013, pp. 171-180.

Digital Object Indentifier 10.1109/MS.2015.26 0740-7459/$26.00 2015 IEEE