

Identifying Suitable Projects For Rapid Development: Some Proposed Selection Criteria

Rashmi Jain, Montclair State University, USA
Anithashree Chandrasekaran, USA
Lymari Castro, Department of Defense, USA

ABSTRACT

The use of rapid development techniques allows building high quality systems in short timeframes while reducing project costs and adding flexibility to the development process so it can be easily adapted to meet customer needs in a volatile market. In order to obtain the benefits of implementing rapid development methods, the development organization and the processes utilized need to be congruent with rapid development methodologies and techniques. A problem encountered by product development organizations is how to identify which projects are suitable for the rapid development approach. An extensive academic research about the implementation of rapid development techniques in Systems Development Life Cycle (SDLC) processes resulted in the identification of selection criteria for the determination of project suitability for rapid development methods. The selection criteria identified by this research are essential elements that must be present in every rapid development environment in order to obtain the benefits of implementing rapid development techniques.

Keywords: Rapid Development; Life Cycle, Selection Criteria, SDLC

INTRODUCTION

To ensure leading the competition in a volatile market, companies have recently adopted the use of rapid development techniques that rely on an evolutionary development approach to build usable systems in short development cycles of 30, 60, or 90 days (Leffingwell, 2007). Rapid development is achieved by implementing a series of techniques that are focused on speed, adaptability to requirements changes, and evolution (Rapley, 1995).

An extensive research of rapid development techniques indicates that the rapid development approach is not suitable for all project types. The suitability of the techniques for a given project depends on the characteristics of the team, the type of project, and its organizational environment (Boehm et al., 2004).

A problem faced by product development organizations is how to identify the suitability of their projects for the implementation of rapid development techniques. This research paper identifies selection criteria for the determination of project suitability for rapid development methodologies and techniques. The selection criteria identified by this research are a set of development techniques and organizational factors that enable organizations to deliver high quality products in short timeframes while having the flexibility to address customer requirements changes throughout the development process in a dynamic environment. The selection criteria provide development organizations a mechanism for the assessment of the likelihood of achieving a faster time-to-market for their products while reducing project costs and achieving a high level of performance.

Selection Criteria For Rapid Development Projects

The authors' research of rapid development methodologies and techniques in SDLC processes resulted in the identification of the unique development techniques and organizational factors that distinguish the difference between rapid and traditional development projects. This set of technical and organizational characteristics constitute the selection criteria for rapid development projects discussed in this paper. The selection criteria are elements or characteristics that must be present throughout product development in order for an organization to achieve a faster time-to-market, lower project costs, and flexibility to address customer requirements in a volatile market.

Criterion #1: Unstable Requirements

The authors' research clearly indicates that requirements volatility is a major characteristic of rapid development projects which are characterized for having dynamic requirements that are driven by the market. The projects have volatile requirements that cannot be pre-specified and are added during the course of the project (Cao et al., 2008). Since volatile requirements can have a negative impact on project schedule and cost, it is important that the development process is flexible enough to evolve and adapt to satisfy new requirements and deliver a working system within budget and schedule. Rapid development techniques and processes provide flexibility to redesign a project in order to limit its exposure to rapid change. In rapid development projects, customers are not required to commit to all functional requirements up front and they can change their minds about functionality as the project progresses. Customers can alter the development priorities so that at every lifecycle stage, the most urgent and critical business needs are met first (Rapley, 1995). The scope of the iterations is kept flexible to allow new requirements to be included throughout the project's lifecycle, but resources and schedule are fixed to ensure the timely delivery of a partial system at the end of an iteration. In rapid development projects, acceptance criteria for testing and deployment requirements are identified up front to ensure meeting customer expectations at the time of delivery and reducing the amount of rework.

The conducted research indicates the following rapid development techniques that can be utilized to manage requirements volatility: identify and include deployment requirements early in the project lifecycle, early planning for scope and project definition changes throughout the lifecycle, and continuous identification and tracking of requirements throughout the project's life (Jain et al., 2011).

Criterion #2: Multi-Skilled Teams

Rapid development organizations rely on highly and diversely skilled individuals that are not specialized in a single technical area (Choperena, 1996; Boehm et al., 2004). In order to build a product in a short timeframe, the organization of the team and the utilization of skills is critical (Butcher, 2007). Multi-skilled teams can be easily assigned to different tasks as needed in order to support the development effort which results in faster development and reduced project costs. The three main functions recommended for a team involved in rapid development activities are development teams, customer proxy, and quality assurance (Elssamadisy, 2008). Teams are comprised of a mix of senior and junior people who are risk takers, results-oriented, and willing to receive feedback on design and delivery. Teams are empowered to make decisions that affect the project with minimum managerial supervision. Teams are encouraged to self-organize and make decisions based on their knowledge and experience. Management encourages development teams to take risks, communicate freely and assume leadership roles (Choperena, 1996). Rapid development projects rely on face-to-face communication as the primary method of knowledge transfer. Therefore, teams are usually collocated. Collocated teams eliminate communication breakdowns, enable rapid customer feedback, enforce collaboration and knowledge sharing, and promote frequent communication that results in the fast resolution of requirements questions (Cao et al., 2008; Teasley et al., 2002). In rapid development projects, teams are organized around physical or logical subsystems. Rapid development organizations should focus their human resources processes on creating teams that exhibit a combination of multiple disciplines and that are creative and innovative (Jain et al., 2011).

Criterion #3: Modularity

Rapid development projects can be broken in short increments. Modularity is essential to reduce project complexity and size and to achieve faster delivery by implementing rapid development techniques such as concurrent engineering, timeboxing, and incremental development. The typical duration of the increments is 30, 60, 90, or 120 days. Increments longer than 120 days could result in extensive designs and schedule slippage (Leffingwell, 2007). Rapid development projects are delivered in short increments that are iterative in nature (Abbas et al., 2008). Each iteration is timeboxed (i.e., fixed resources and schedule) and is comprised of the development lifecycle processes of requirements, design, implementation, testing, and deployment (Jalote et al., 2004). In the first evolution of a rapid development project, the initial focus is placed on breaking the system in short increments; the second evolution overlaps in a phased approach; and the third evolution shortens the phases into iterations that are focused on value to the customer (Cottmeyer, 2008). Rapid development projects should be planned for an increased number of release cycles that include customer approved value and added functionality (Jain et al., 2011).

Criterion #4: Continuous Customer Collaboration

Rapid development projects are successful in meeting customer requirements in a volatile market by incorporating the customer as an integral element of the development process (Jain et al., 2011). In rapid development projects, customer requirements drive the development effort and testing; therefore, the customer is involved during all phases of project development. This allows early identification of new requirements in a changing market. Rapid development projects can only succeed if the customer works with the development organization to describe, prioritize, and refine their needs throughout the development process. In order to achieve this, a customer representative is assigned to the development organization. The customer representative needs to be knowledgeable of the project and the market, be authorized to make decisions that can impact the project, be collaborative with the development organization, and committed to the project (Boehm et al., 2004). This will aid the development organization to make informed decisions about new requirements, test cases, and acceptance criteria for the project. Late customer involvement could result in the delivery of systems that do not meet customer needs at the time of delivery because customer requirements and business needs could have fundamentally changed.

Criterion #5: Frequent Delivery (I.E. Iteration Prototypes)

Rapid development projects are delivered in short increments that are developed using an iterative approach (Abbas et al., 2008). Each iteration provides a partial version of the system under development (i.e., prototype). Prototypes allow obtaining quick customer feedback on system requirements. They are used as a mechanism to formulate and evaluate requirements, specifications, and designs (Luqi et al., 1992). They can be as simple as graphical representations of the system or a fully operational version of the system. Frequent delivery of functionality can significantly reduce project costs since it constrains most of the fixes to a small module of the system. The requirements of each increment are continuously tested and integrated until an iteration prototype is ready for delivery to the customer. Each increment delivers the highest priority requirements to the customer. The next increment of the system builds upon the functionality and requirements of previous iterations and incorporates the new requirements with previous requirements that are prioritized by the customer and the development team. Therefore, each delivery of the system is incremental and has more functionality than the previous iteration. The use of iteration prototypes can significantly reduce project costs since it allows the early identification of defects and the identification of new requirements. The technique of frequent delivery of prototypes adds flexibility to the development process by allowing the best architectures to evolve iteratively (Jain et al., 2011).

Criterion #6: Adaptive Planning

Project planning in a rapid development environment is an iterative and incremental process that continuously adapts as the developers and customers knowledge of the system increases (Hass, 2007). Developers and project stakeholders work together to plan each iteration using information learned from previous iterations and prioritizing iteration requirements based on their risk level and business value. The requirements with the highest risk level and business value have higher priority. Since the scope of the project remains flexible in rapid development projects, resources and schedule are fixed before work starts on a new iteration so there is more control

during project development. Project managers are involved early in the project's life so problems can be detected when they are small and easy to solve. The management team is continuously informed about project progress by having frequent informal meetings. Management shows interest in project progress and encourages personnel characteristics such as empowerment, independence and leadership skills to emerge within teams. Project plans are developed and updated continuously throughout the project's lifecycle to manage requirements volatility (Jain et al., 2011).

Criterion #7: Continuous Testing And Integration

Rapid development projects achieve fast time-to-market and high quality by incorporating testing and integration as a continuous process throughout the development lifecycle. This technique provides stability throughout the project's life by constantly evaluating the architecture, interfaces and implementation, and by preventing the accumulation of latent defects (Forte, 1997). Iteration prototypes enable quick customer feedback that is used to validate the system and plan future iterations. In rapid development projects, potential risks are identified early in the project lifecycle to evaluate and analyze designs (Jain et al., 2011).

Criterion #8: Continuous Risk Management

Due to the volatility of the target market, rapid development projects need to continuously assess the risk of not meeting customer expectations on time and with the desired level of quality. Each iteration of a rapid development project is planned by assessing the risk of the features (i.e., requirements) to be developed. The features with the highest risk are developed in early iterations of the system to reduce the likelihood of not meeting the proposed schedule (Leffingwell, 2007; Smith et al., 2005). Continuous risk management is achieved by performing continuous testing and integration and reviewing iteration prototypes at the end of each iteration to verify the system meets the acceptance criteria. In rapid development projects, the continuous risk management process involves not only assessing the risk level of the features to be implemented, but also managing the risks associated with the team and the development processes. In rapid development projects, continuous risk assessment is utilized to guide the decision-making process. In rapid development, the risk management process starts with the identification of relevant checkpoints that are used to assess the risk level throughout the lifecycle (Jain et al., 2011).

Criterion #9: Continuous Verification And Validation

Verification and validation in rapid development projects is an iterative and concurrent process that is conducted throughout the system's lifecycle. A continuous verification and validation process allows to identify defects in a timely manner and contain them in small modules, thus reducing the cost associated with rework (Leffingwell, 2007). The frequent delivery of functionality combined with continuous customer feedback provides a constant validation and verification process in rapid development projects (Jain et al., 2011).

Criterion #10: Low External Dependencies

In rapid development projects, the customer drives the requirements. Therefore, projects tend to be stand-alone systems that have low dependencies on external systems. The rapid development approach is not suitable for projects that have many interfaces with COTS or legacy systems because these tend to determine the requirements (Boehm, 2000; Turner, 2004) which could have a negative impact on project schedule and budget. The use of less complex architectures can also speed up the development process. Rapid development projects tend to utilize mature technologies (Jain et al., 2011) for which performance and reliability is already known. By minimizing the use of new technologies and dependencies on external technologies, rapid development projects can reduce the uncertainty introduced by novel technologies and COTS vendors, which increases the likelihood of meeting project schedule and budget.

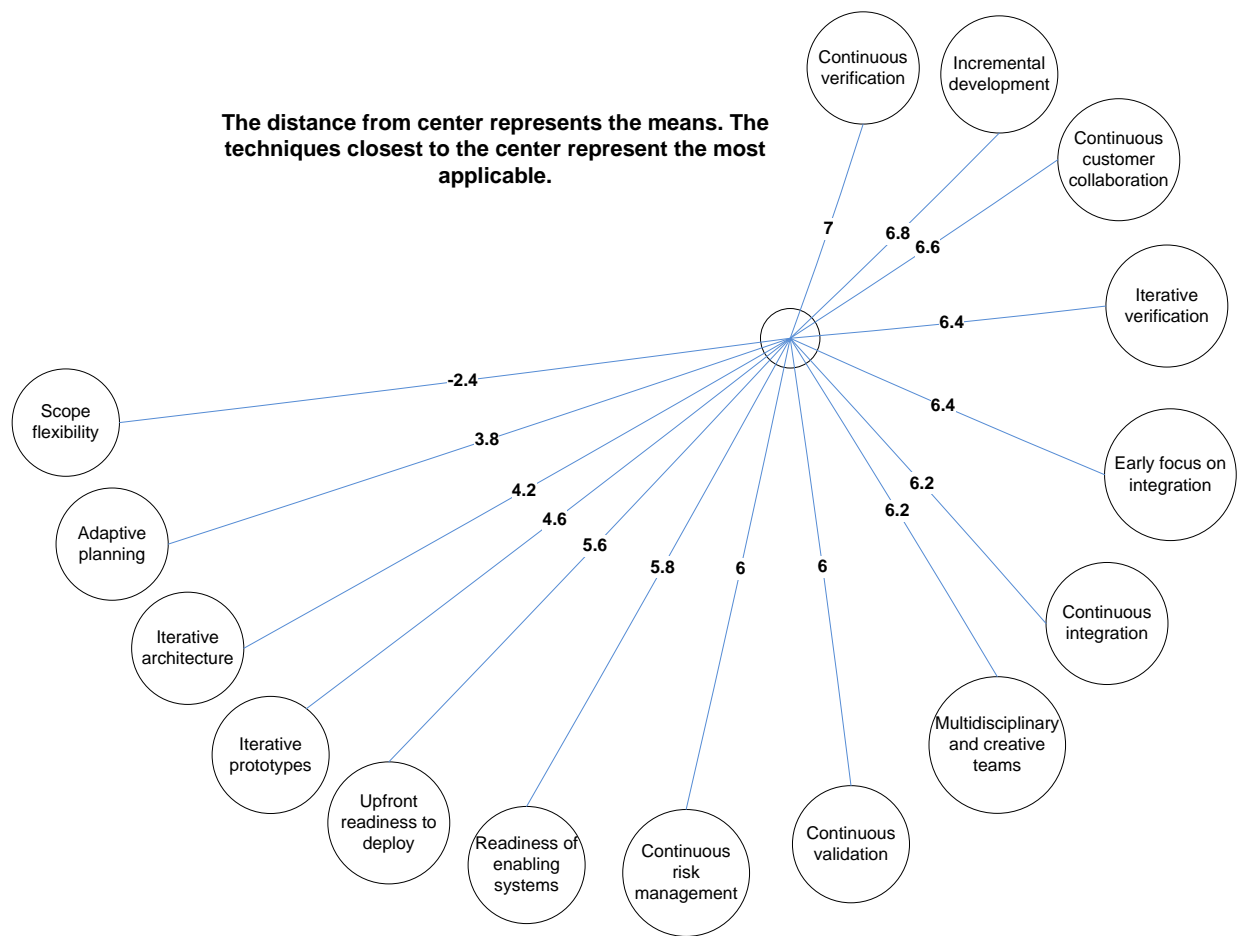
RESEARCH METHOD

The research included of an extensive literature review of rapid development methodologies. A survey methodology was utilized to study the applicability of rapid development techniques to SDLC processes. The survey

was comprised of 22 questions in total. The survey was utilized to collect data from a population comprised of program managers and systems engineers from both the commercial and defense sectors. A 5-point Likert scale was utilized to measure the level of agreement with the various rapid development techniques. Respondents’ experiences ranged from 9 to 31 years in the creation and implementation of systems in their industry.

RESEARCH ANALYSIS AND RESULTS

By mapping the rapid development techniques with the various selection criteria, it was possible to determine the level of importance of the criteria as perceived by the surveyed population. The mean and standard deviation were calculated using weighted averages (-2 to 2 for Strongly Disagree to Strongly Agree). Figure 1 presents the level of agreement to the techniques where the closer to the center, the higher the agreement and the higher mean score. The results clearly indicate a high level of agreement with the rapid development techniques and provide a strong validation for the selection criteria.



Adapted from Jain et al., 2011
Figure 1: Survey Results Showing The Mean Of Rapid Development Techniques

Table 1 shows the mapping between rapid development techniques (Jain et al., 2011) and the selection criteria.

Table 1: Mapping Between Rapid Development Techniques And Selection Criteria

Selection Criteria	Rapid Development Techniques
Unstable requirements	Scope flexibility Upfront readiness to deploy
Multi-skilled teams	Multidisciplinary and creative teams
Modularity	Iterative and incremental development
Continuous customer collaboration	Continuous customer collaboration
Frequent delivery (i.e. prototypes)	Iterative prototypes Iterative architecture
Adaptive planning	Continuous and adaptive planning
Continuous testing and integration	Early focus on integration Continuous integration
Continuous risk management	Continuous risk management
Continuous verification and validation	Continuous verification Continuous validation Iterative verification
Low external dependencies	Readiness of enabling systems

Figure 2 illustrates the level of importance that the respondents assigned to the various selection criteria for rapid development projects. These results indicate that modularity is the most relevant criteria in the selection of rapid development projects. This indicates the importance that respondents assign to decomposition of a project into short increments to reduce its complexity and be able to implement rapid development techniques such as timeboxing, concurrent engineering, and iterative development. Continuous customer collaboration was identified as another highly relevant criterion. This is an indication of the positive impact customer involvement has on steering the development process in a direction where schedule, budget, and performance can be met. The results show that being able to conduct the concurrent engineering activities of continuous verification and validation and continuous test and integration is relevant for the selection of rapid development projects. The use of multi-skilled teams was identified as being relevant for the selection of rapid development projects. The continuous risk management criterion had a high level of relevance for the selection of projects for rapid development which indicates the need of assessing risks at all times during the project's lifecycle in order to make informed decisions.

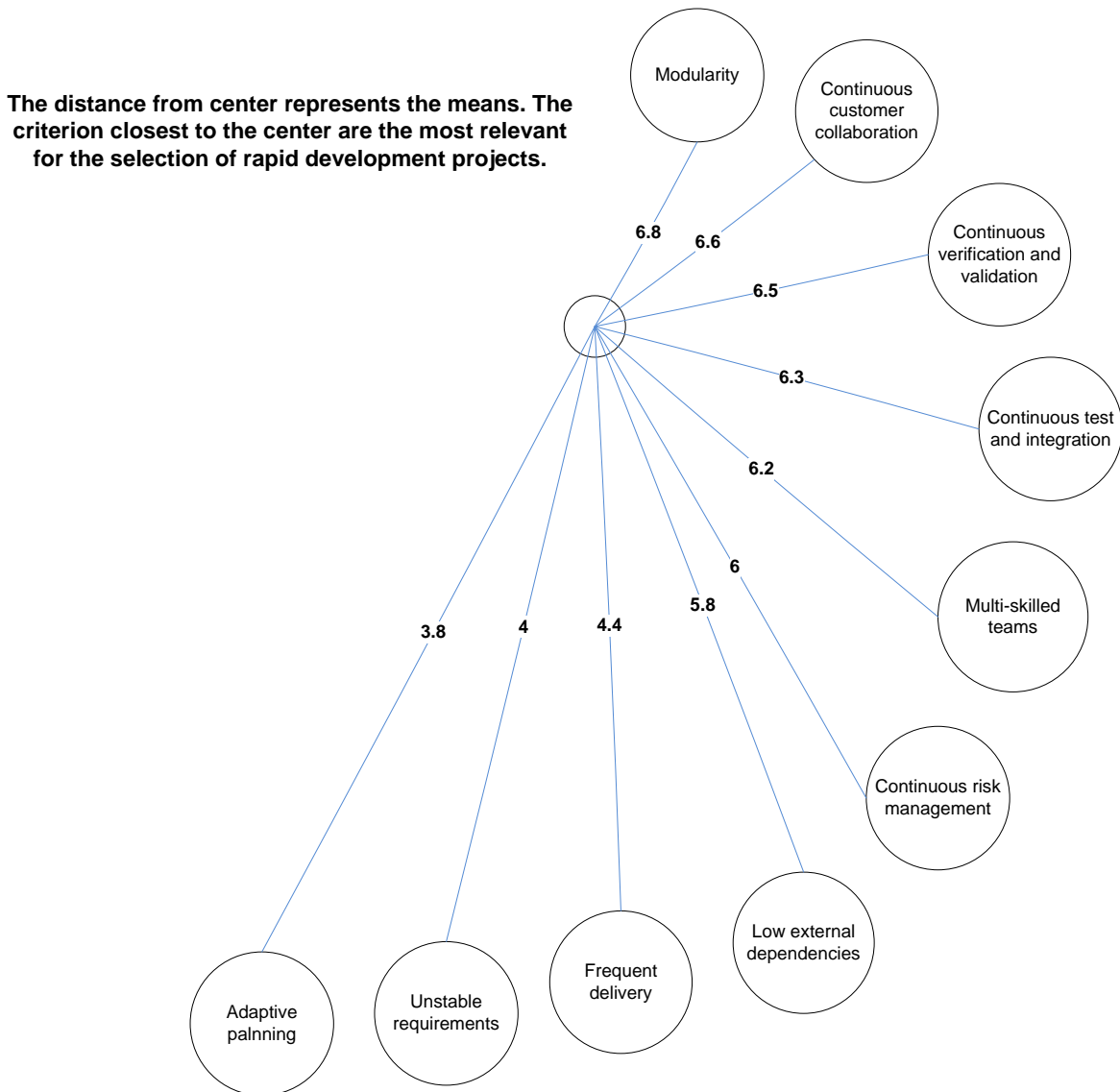


Figure 2: Survey Results Showing The Level Of Importance Of The Various Selection Criteria

The four selection criteria identified as least relevant were low external dependencies, frequent delivery, unstable requirements, and adaptive planning. This could be an indication of several factors:

- In projects where there are many external dependencies, continuous risk management helps identify and mitigate the risks associated with the external systems before they have a negative impact on the project; therefore, having low external dependencies is not seen as a strong selection criterion in projects where a precise continuous risk management process has been established.
- Although frequent delivery is important to manage requirements volatility, it is not always feasible to deliver a fully operational prototype at the end of an iteration.
- Although rapid development projects are characterized for having unstable requirements, the methodologies are fully applicable to projects that have stable requirements. The low relevance of the adaptive planning criterion could be a consequence of the diverse background of the surveyed population where the planning process is conducted differently in the commercial and defense sectors due to the nature of their projects.

CONCLUSIONS

The results of the survey clearly indicate the importance of selecting projects by taking into account both the technical aspects (i.e., continuous test and integration, continuous risk management, etc.) and the organizational characteristics (i.e., multi-skilled teams, continuous customer collaboration) of the project to ensure obtaining the benefits of implementing rapid development techniques. This research will be followed by a more in-depth analysis of the impact of the selection criteria on project cost, schedule, and performance.

AUTHOR INFORMATION

Dr. Jain is currently Professor and Chair for the Department of Information and Operations Management at Montclair State University in New Jersey, USA. She has extensive experience of teaching graduate students and senior executives both in the U.S. and overseas. Some of these are Defense Science and Technology Agency (DSTA) Singapore, FAA USA, IBM, Japan Aerospace Exploration Agency (JAXA), etc. She holds a M.Sc. and Ph.D. in Technology Management from Stevens Institute of Technology, USA. Her research focuses on systems design and integration, rapid development, business analytics etc. Dr. Jain is on the editorial boards of several international journals and is on the technical program committees of various international conferences. She is a visiting professor to the Graduate School of System Design and Management, Keio University, Japan.

Anithashree Chandrasekaran is an IT Risk, Security, and Compliance Manager. She has her PhD in Rapid Systems Development and its impact on Project Risks from Stevens Institute of Technology, New Jersey. Primary focus of her academic research has been in rapid systems development and the risk associated with such development. Her other focus areas include Application Security and Risk, IT Compliance, Agile SDLC, Requirements Engineering, Systems Architecture and testing. She is part of the IT industry in the Insurance and Financial Services sector.

Lymari Castro is a Technical Director for the Department of Defense (DoD). While in DoD, she has been a Lead Systems Engineer (SE) and Software Development Team Lead for DoD Major Systems Acquisition Programs where she successfully implemented SE agile systems engineering processes blended with an overarching traditional waterfall methodology. She previously worked as a Radar Systems Engineer at Raytheon. She has a MENG in Systems Engineering from Stevens Institute of Technology, MENG in Engineering Physics from Cornell University, and B.S. in Physics from University of Puerto Rico. Her research interests are rapid systems development and systems integration.

REFERENCES

- Abbas, N., Gravell, A., & Wills, G. (2008). 'Historical Roots of Agile Methods: Where Did "Agile Thinking" Come From?', *Lecture Notes in Business Information Processing*, vol. 9, pp. 94-103.
- Boehm, B. (2000). 'Requirements that Handle IKIWISI, COTS, and Rapid Change', *IEEE Computer- Software Management*, July, pp. 99-102.
- Boehm, B., & Turner, R. (2004). *Balancing Agility and Discipline: A Guide for the Perplexed*, Boston, MA, Addison-Wesley.
- Butcher, T. (2007). 'Supply Chain Knowledge Work: Should We Restructure the Workforce for Improved Agility', *International Journal of Agile Systems and Management*, vol. 2, no. 4, pp. 376-392.
- Cao, L., & Ramesh, B. (2008). 'Agile Requirements Engineering Practices: An Empirical Study', *IEEE Software*, vol. January/February pp. 60-67.
- Choperena, A. M. (1996). 'Fast Cycle Time- Driver of Innovation and Quality', *Research Technology Management*, vol. 39, no. 3, pp. 36-40.
- Cottmeyer, M. (2008). *The Evolution of a Project Schedule: Agile Development Practices*. Orlando, FL.
- Elssamadisy, A. (2008). *Self Organizing Teams. Agile Adoption Patterns: A Roadmap to Organizational Success*. 1st ed. Boston, MA, Addison-Wesley.
- Forte, G. (1997). 'Managing Change for Rapid Development', *IEEE Software* vol. 14, no. 2, pp. 120-122.
- Hass, K. (2007). 'The Blending of Traditional and Agile Project Management', *PM World Today*, vol. IX, no. V, pp. 1-8.

- Jain, R., Chandrasekaran, A., Castro, L., & VanLeer, M. (2011). Feasibility of a Rapid Systems Engineering (RSE) Framework: An Exploratory Study. *International Journal of Industrial and Systems Engineering*, 7(1), 45-65.
- Jalote, P., Palit, A., & Peethamber, V. T. (2004). 'Timeboxing: a process model for iterative software development', *The Journal of Systems and Software*, vol. 70, pp. 117-127.
- Leffingwell, D. (2007). *Scaling Software Agility: Best Practices for Large Enterprises*, Boston, MA, Addison Wesley Professional.
- Luqi & Steigerwald, R. (1992). 'Rapid Software Prototyping', *IEEE Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, vol. 2, no. 7-10 Jan, pp. 470-479.
- Rapley, K. (1995). 'RAD or TRAD or Both? The Future of Software Development', *IEEE: IEE Colloquium on Will TickIT and ISO 9000 Survive Rapid Application Development?*, 11 Dec., pp. 1-3.
- Smith, P., & Pichler, R. (2005). 'Agile Risks/Agile Rewards', *Project Management*, Software Development, April, pp. 50-53.
- Teasley, S., Covi, L., Krishnan, M., & Olson, J. (2002). 'Rapid Software Development Through Team Collocation', *IEEE Transactions On Software Engineering* vol. 28, no. 7, pp. 671-683.
- Turner, R. (2004). 'Toward Agile Systems Engineering Processes', *CrossTalk*, April, pp.11-15.

NOTES