

PRODUCT PERSONIFICATION: PARAG MODEL TO SUCCESSFUL SOFTWARE PRODUCT DEVELOPMENT

Sriramasundararajan Rajagopalan¹
¹Agile Training Champions, USA

ABSTRACT

The ubiquity and abundance of web, mobile, hybrid, and wearable software in today's technology driven lifestyle is unmistakable in personal, social, and professional life highlighting the importance of software product development. With the growth of agile and lean approaches to product development, the responsibilities associated with many of the traditional roles have merged and new roles emerging. The impact of these new trends on software product development have led to misunderstandings leading to product owners bringing their own personal approaches to product development failing to focus on the actual product itself and the effect on the performing organization, clients, and end user. This paper focuses on a new paradigm shifting the focus from product owner to the software product itself by giving it a life and voice to echo the characteristics needed for its adoption, penetration, and sustenance.

KEYWORDS

Product Development, Product Management, Software Product Development, Traditional Product Management, Agile Product Development, Product Persona, Project Management, Program Management, Portfolio Management, PARAG Persona, Persona, Successful Product Attributes

1. INTRODUCTION

Software product development has come a very long way. Emerging from the keyboard input based command driven menus in the mainframe, UNIX, and Disk Operating System (DOS) world, the software products have become very sophisticated evolving into the web-centric and mobile devices. The grand approaches to product development, focused approach to product innovation, and process oriented approach to the relationship between management and development processes [1] have had pronounced effect on software products.

For example, consider living without the following software products and their associated benefits:

- Mobile apps on the smart phones and portable tablets connecting with the family and friends globally through social networking and communication tools
- Satellite enabled geo-targeted business-to-business and business-to-consumer applications
- Global positioning system in automobiles
- Smart applications enabled televisions and house monitoring systems
- Secured and encrypted access to work email, documents, corporate applications, meeting places and distributed virtual collaborative tools
- Wearable technology devices beyond personal fitness devices and smart watches to critical timely healthcare assistance aids

It is unequivocally obvious that the software product development has gone through a roller coaster drive on creative approaches to problem solving using technological innovation,

leveraging economies of scale in resource utilization and talent maximization [2], and benefitting from the organizational level innovation in improving efficiency through lean approaches to reducing cost of production, eliminating waste, and improving the total cost of ownership [3]. The challenges of software product development takes on a special significance as these software applications pose several unique challenges such as the multitier architecture separating business, application, and data layers, security and privacy restrictions, compliance and audit limitations, limited real-estate constraints in the mobile and tablet devices, multiple screen orientation challenges, networking considerations, usability and continuous updates among them.

This sophistication in the software development emphasizes that the product manager become multifaceted to understand the technical, operational, and economic feasibility of the product development often expected in a business systems analyst. Cohn [4] further brings focus into the agile approaches to software product development on the product owner role merging additional responsibilities differentiating the product manager in a traditional software development from the product owner in the agile software product development. Synthesizing these responsibilities of a product manager and product owner across the product life cycle stages of idea generation, feasibility analysis, design, development, build, marketing, validation, sustainability, and retirement, Rajagopalan [5] emphasizes several important characteristics of the product manager integral to product development.

Brown [6] assimilates copious research on the importance of the dynamics of the team that worked on developing the product, the product owner's personal style, or the project manager's execution efficiency in product development. Subsequently, Tau [7] observes where deviations from these norms may unleash several *faux pas* in product development. Some examples may include the following:

- Right product being developed with the wrong features due to not properly managing the stakeholders
- Miscommunication due to the additional layers of customer facing roles leading to the development of non-value added features delaying the launches
- Customer's failure to accept the product due to its fitness for continued use being no longer viable in the market place due to poor features, performance, and antiquated technology
- Inadequate positioning of the product marketing in its pre- and post-launch preparations
- Not evaluating the sustainability of the product to be profitable for the organization for continued investment
- Incomplete market research conducted to initiate new product development or prolong the existing product despite its declining market share

Nevertheless, when consumers purchased any commercial off the shelf (COTS) software, leased software as a service (SaaS) or platform as a service (PaaS), or downloaded a mobile application, are their choice of the product driven by the agile development practice used in product development or is their preference determined by the attributes of the software or the platform, such as its price, usability, stability, performance, quality, fitness for continued use, and supporting services? According to Gorchels [8], the organization's credibility spearheads in the form of brand equity determining the product's value proposition leading to the customer's buying decisions. While that is generally acceptable, several new small firms' boasts successes [9] without any brand image but with niche features. Therefore, even when a product manager or product owner exhibits all the required characteristics espoused by scholars and practitioners, focusing from a consumer's point of view, it can be deduced that the choice to buy a product is determined by the product's features and not the product manager's style.

Regardless of the development process (plan driven, agile, hybrid, v-model) adopted, if the product owners and marketing managers focused on the product's attributes correctly and allowed customers to experience the products, then the likelihood of the product's adoption and user's engagement with the product will increase leading to the high acceptance in the target market. For example, consider the Kiosk marketing strategies [9] by providers like Samsung, Microsoft, and Apple in their showrooms where end consumers can experiment the software. Other major enterprise and industry specific applications follow these trails in the customer conferences and trade shows. Word of mouth marketing yields substantial advertising because the consumers market the product themselves creating the brand equity and the brand loyalty.

Alan Cooper, known for the iconic user centric design in software development, introduced the notion of a fictitious character with specific traits called as Persona [10] embraced heavily in the agile software development [4]. This persona defines the target customer, their characteristics, hot buttons, success definitions, and acceptance criteria to name a few. Although this persona was originally conceived for an external user, the internal stakeholders have been personified [11]. Cleland-Huang extended the same approach to "architecturally savvy personas" [12] defining the system architecture to elicit requirements for capturing performance, reliability, security, scalability, and other quality considerations to more efficiently and effectively design the software systems supporting the product.

When the customers that use the product and the architecture underlying the systems can be abstracted as persona to provide the guidance to software product development, then this paper questions if there is a persona for the product itself? For instance, if the product had a voice and spoke about the traits that were built into it for its use in the field, acceptance in the market, challenges it faces, and its position within the performing organizations' portfolio, then what would the product say? If such a persona were defined for a generic software application product or customer interfacing platform, what would be the omnipresent characteristics of all such products that are independent of the team, product owner style, and product development practices?

2. PRODUCT PERSONA: A FRAMEWORK FOR PRODUCT DEVELOPMENT

Several books, articles, and best practices [11, 12, 13, 14, 15, 16] discuss numerous requirements in the product lifecycle like the market research, feasibility analysis, target market positioning, pricing and promotion, rollout, training, etc. As the performing organization must be sustaining the product development for the initial acceptance through the concept development and prototype development stages to persist amidst the competitive demands through the clarification and stabilization phases of the Product Development Alchemist framework [17], a few key attributes defining the person of all generic software product can be synthesized.

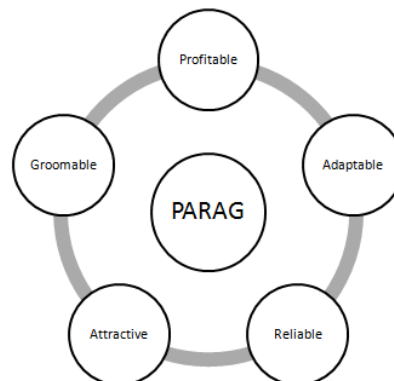


Figure 1. PARAG Product Persona

Summarizing these attributes as illustrated in Figure 1, this paper introduces the PARAG Persona to software product development ensuring that the software development invokes critical thinking on the integration of management and engineering principles and using appropriate key performance indicators to measure these attributes during the product development cycle.

2.1. Profitability Trait

Both the scholar and practitioner worlds relate to the need to price the product appropriately [9]. However, pricing for the software developed is challenging because the intangible cost of building and maintaining the software is often perceived as cost of doing business. Even in an agile setting where customer collaboration is weighed more than contract negotiation [4], Rajagopalan [18] emphasizes the need for features built over many sprints justify the product development function with either the increased revenues or enhanced customer market penetration for the performing organization. Using several anecdotal examples employing agile principles to product development and project management, Schwaber [19] draws the similar foundation for the product owner to use risk adjusted product backlog as a vehicle to maximize the value as part of the return on investment (ROI).

In this context, the “profitability” trait of PARAG persona will expect the product management and other business stakeholders to evaluate the following critical thoughts [17] to ensure that the tangible and intangible organizational resources spent on the product development and custom project management justify the opportunity cost of working on these initiatives throughout the product life cycle.

- Is the product idea valid within the portfolio to create or sustain competitive advantage?
- Has similar products been attempted before and what lessons have been learned not to repeat the failure maximizing strengths and mitigating risks?
- How will the product be monetized for the performing organization or build social branding within its portfolio of products?
- What types of external customer polling, surveys, interviews, and focused assessments been conducted to identify value added features that may position the product for successful adoption and penetration? Has a beta customer been identified?
- What assessments have been done for the internal business units and departments to alleviate their challenges or plan for capacity to support the projects?
- Is the market share substantially large enough to maintain growth? If the market share is small, what is the niche for this product to create a return on investment?
- What is the payback period so that investment in the capital and operating expenses justify the opportunity cost of selecting this product?
- What demographics of potential buyers and their preferences make this product a must to have?
- What are the strengths and weaknesses of the competition that has to be weighed in prioritizing value added features and product rollout strategies to create the market attraction?
- What are the intellectual property issues?
- How long can the initial expectation for the market be before new features are expected?
- What is the timeline for the introduction to market, operationalization of product roll out internally to all the supporting stakeholders, post-rollout assessment strategies?
- What are the exit criteria to abandon further development?

Depending upon whether the performing organization is a price leader or a price follower, the pricing policy may have to factor ongoing maintenance, customer support, rebates and discounts, and the capital affordable to penetrate the market and survive the competition with the growth of

the open source paradigms and low cost differentiators providing cost effective alternatives [4,9]. Nevertheless, if the organization has to sustain the product, then product owner must have conscientious attentiveness towards margin tracking for software development so that the organization remains profitable [16, 19]. Therefore, profitability for the performing organization is an important product persona trait that the product owners and managers should be consciously aware of so that the resources are utilized for developing the right features not only retaining the existing customers but also gaining new customers through increased engagement.

To satisfy this profitability focus of PARAG persona, at a macro level, the product owners can utilize the risk adjusted expected commercial value (ECV) [16] by factoring the costs of development and costs of product roll out and launches along with the probability of technical success followed by the probability of commercial success. Additionally, at a micro level, the product owners may use burn rate as an effective measure to see how much of their budget allocated is being consumed against the revenue generated. The number of hours allocated to a specific set of product development features can be evaluated using the earned value metrics (EVM) like the cost performance index (CPI) recommended by Project Management Book of Knowledge [20]. Furthermore, the product managers and owners can effectively use these metrics along with the cost of delay associated with the wait times [21] to evaluate the risk versus reward increasing the customer value while remaining profitable.

2.2. Adaptability Trait

Even if one product enjoys the first mover economic advantage, if it chooses to not adapt to the market trends, it may lose to the competition. Look at how Motorola lost its market share to Nokia & Erickson by not adopting the digital cellular trends [17, 22]. The operating environments of a software product also are continuously subject to changes as the characteristics defining the target population, federal mandates, and technological backdrops may change. As it is far difficult to get a new customer, it is imperative to constantly assess the refinements in the product while continuously improve the product with additional features staying ahead of the competition.

Williams [23] recounts the client's concerns on the dedicated account executives just responding to the client's requests less valuable and citing that as a rationale for switching to a different agency. Rajagopalan's summary [24] of the career progression framework for the account management and project management professions to evolve as product owners in the agile setting accentuates the importance of product and domain knowledge as critical pillars to being adaptable to perceive the unsaid and said expectations of the customers [9] advocated by the TONE framework [25]. "The agency never gave anything we didn't ask for," summarizes Williams (para 7) inferring that the source of new and expanded software features is not always from the client [23] but expected of client facing roles in today's flat hierarchical structures.

It is in this context the "adaptability" trait of PARAG persona highlights the inevitable significance of the client facing roles in custom project management, program management, account management, product development, marketing management, and portfolio management. When these roles become attuned to the voices of several internal stakeholders, external vendors, suppliers, and clients, the ground becomes fertile for creative ideas to develop for the internal teams and external clients to continuously improve and symbiotically collaborate to extend the software product to exceed demands. Bergeron and Blander [17] distinguishes this stage as the clarification stage in their Alchemist model of product development where the beta customers strive to become the actual product users co-creating the value proposition story [26].

The “adaptability” trait of PARAG product persona will demand the product management and customer facing roles to evaluate the product after its inception in multiple angles from both the technology and the market focus.

- What types of technology assumptions are made on the software development tools and platform that could invalidate the product to lose its market share and sustain growth in the near future?
- What types of additional end user applications for internal business units and external clients will be required for maximizing the adoption and penetration without additional new enhancements?
- What requirements exist for beta customers to be converted to actual customers to foster engagement?
- Are there competing technologies on the horizon that could jeopardize the market share for the product?
- What types of end-user training and documentation will become inexorably essential for continued engagement and use of the products?

Evaluating the types of metrics for measuring adaptability is particularly challenging because of the inherent subjectivity in its definition. Subramanian and Chung [27] come to the rescue in defining adaptability as the extent to which a software system can adapt to the changes in its environment without any external intervention. For instance, although a user story or requirement for a webservices to connect and exchange data adds value to the customer, the product owner should encourage and insist on business value added user stories or requirements to monitor the health of the system by screening the event logs for specific keywords or error codes. Such a proactive alert automatically raised for a database or network connection timing out, for example, exemplifies adaptability.

Subramanian and [27] Chung extend that this adaptability can be measured at four different levels in a software product namely the conceptual, modular, execution, and code architectures as part of a process oriented non-functional requirements (NFR) framework [28] to measure software adaptability. These types of user stories to develop the health monitors as mentioned before are similar to the various bioelectrical signals like the visible skin colour, blood pressure, temperature, or pulse in the human body before involving deeper blood tests for subsequent analysis. The product owners can then evaluate how much portion of the release cycles have such business value added requirements for production monitoring to ensure product health and evaluate that as part of the burn down charts.

2.3. Reliability Trait

One of basic tenets of agile product development [4, 19, 29, 30] is the frequent release of minimally marketable features to the client. Imagine a scenario where every product release is either having increased escaped defects to be fixed in subsequent release making the product unstable. Then, the customers trust is eroded. While it is possible to properly architect software to safely land the Curiosity on the surface of Mars [31], there were many reliability questions in the nationally acclaimed federal healthcare site rollout to register for healthcare coverage [32]. In today’s digital workforce dominated by impulse for people to blog, tweet, or comment about the product features, the marketability rating cascades exponentially down if the product is not reliable.

Software developers, system architects, and business analysts may often think in terms of the modular code [27] that when properly tested would increase reliability [33]. While that is true [30], even when not extending the core agile principles [4, 19, 34] to reliable software product,

the “reliability” trait of PARAG persona would incorporate the following thoughts to ensure the reliability both from the customer’s and business value perspective.

- How much is the software optimizing the use and monitoring of the system’s resources [27, 30, 33] like memory, storage, and CPU power in client-installed, web-hosted, and mobile products?
- How intuitive and user friendly is the graphical user interface [35] using widgets, touch sensitive screens, physical motion, and virtual keyboards instead of the usual windows, icons, menus, and pointers, and workflow [36, 37] considering the increasing real estate area limitations, screen orientation, and wireless capabilities on the mobile environments?
- What processes like requirements traceability, acceptance testing, automated testing, and refactoring are utilized, reviewed, and followed to focus on both conformance to requirements and fitness for use [38] without any defects created due to requirements misses, engineering slips, and QA omissions?
- What types of audits are consistently done to protect data adequately [39] against the malignant users using network intrusion, SQL injection, malware, and accidental loss of unencrypted data in the portable and mobile devices?
- How effectively simple are the solutions designed and implemented so that the architecture is not too complex with numerous alternative paths [40] lending the solution inadequate for adequate automation leading to defects?
- How engaged, involved, and enforced are the acceptance testing from a product, business, and client perspectives [41, 42] before rolling out the software to end users?

Several types of metrics can measure these above elements [43]. For instance, for an object oriented programming environment, the depth of inheritance, the signature of the function calls, and the complexity of the paths are all examples. In the database environment, the level of normalization applied on the database and the role based privileges for data integrity and security are examples to monitor during development. On the network side, many types of infrastructure usage can be monitored proactively, such as the memory or CPU usage, processes or frequency and types of information written to transaction or event logs and heartbeat monitors to monitor the events in these logs, use of the network ports, service status of various libraries. Graphical user interface can evaluate the use of Ribbon interface to condense and configure menus, navigation, and toolbars, use of graphs with options to view on a bigger scale, on-screen help with minimal click, etc. [44]

2.4. Attractive Trait

Unlike many of the brick-and-mortar showrooms where tangible products can be seen, touched, and felt before purchase [9], the software products are intangible. For instance, Google’s prominence in the software industry is its proprietary focus on search engine optimization. The consumers don’t relate to the detailed SEO optimization guidelines but only to the results. Customers that get the software products, desktop or mobile devices, and other peripherals are becoming more savvy looking for easy access to get the product by downloading the installable package, simplicity of installation and un-installation, configurable customization and localization, feedback mechanism, rating preferences, evaluation editions, and many others making the software attractive with copious differentiation strategies essential in positioning the product for success [45].

While the above areas of giving the showroom presence besides the convenience of installation, access to upgrades, and gathering feedback are indications of software product attractiveness, there are other areas of product attractiveness. Ossadnik and Kaspar [46] observe that the

software product attractiveness is a measure of how seamlessly the product's inherent capability can be understood, learned, and used [46]. For instance, very little professional training institutions or educational institutions offer training on how to use social networking sites like Twitter, Facebook, LinkedIn, or Instagram. Similarly, the commercial software applications make self-learning *sine qua non* of their software products. Consider tools like Camtasia screen editing application or Version One agile product development tool to name a few. These software suites rollout a "How-To" video that minimize the learning curve associated with learning these software products making them attractive to the end users.

Urban and Hauser [47] succinctly summarized the attractive product features should go beyond just meeting the customer needs and addressing their unsaid needs [9] better than the competitors. For example, attempting to comprehend how the product is being perceived and the extent to which it can be self-taught will help the product owner to adjust the prioritization of the product backlog so that the product becomes a choice that the customer cannot live without. The lack of a good technical knowledge and business acumen in the client facing roles will inescapably determine how much of these unsaid needs can be inadequately comprehended [23, 25]. The agile settings address this gap by making the customer part of the extended agile team [40].

Say, we narrow the focus to mobile application development. Wasserman [35] gives scores of challenges for mobile application development that may make the software product very unattractive, such as the high likelihood of a mobile application failing to factor the limited capacity of the battery storage parasitically draining its power to attract negative ratings. Similarly, while mobile optimized websites may be attractive, these responsive designs will gain less favourable ratings from users who can't take advantage of the full web functionality on the smaller mobile devices [35].

If these challenges are not adequate, the need for mobile applications to be developed independently for iOS and Android users demand further development considerations. Consider as an example one of the popularly known application, Trello, that is used as a Kanban tool. Besides simulating the simplicity of tasks on a Kanban task board, the tool provides customization of the sticky notes, notification alerts, deadlines, and membership to the board among its features. Most importantly, the tool provides instant access to the same information through web, iOS, and Android with a cool interface making it easy to learn underscoring the need to have the code architecture that needs to be carefully weighed for multiple environments. These challenges underpin the increasing demand for different creative thinking and performance considerations for the mobile application from their web counterparts while still maintaining the data integrity between the two modes.

Therefore, carefully positioning and differentiating the product [9] attractively for the target market with value added features [45, 46, 47] that not only augment its initial rapid adoption but also its continued engaging use besides the ability to communicate incremental updates penetrating the markets both vertically and horizontally is becoming the inevitable characteristics that the "Attractiveness" trait of PARAG persona will seek in the product development and business management roles [48]. So, software products should definitely focus on a few metrics like the customer perceived value and the total customer lifetime value that varies based on the type of software developed. A few thoughts to keep in mind to see if the product is attractive are:

- How efficient is the product marketing during its pre-launch and post-launch stages?
- What is the effectiveness of the marketing channels and what is the return on investment on efforts expensed versus the engagement generated?

- How does the customer associate with the product beyond the product's core identity justifying the lifetime value of the product for the customer?
- If the customers can live without the product due to competition, what features are critical to make them return to engage with the product?
- What areas are the targeted consumers watching for improvements, like geo-targeting, wifi enabled printing, data integrity between apps, and the availability of apps on many device types, that determine if the investment in our technology needs to be evaluated?

2.5. Groomability Trait

Regardless of how great a product is, if the competitive five forces [9, 49], such as the threat of intense rivalry, new entrants, substitute products, buyer's growing bargaining power, and suppliers growing bargaining power are not constantly monitored, then any great product loses its reputation. The "groomability" trait of PARAG persona is related to the "attractiveness" trait but takes a special focus as the product is relatively past its growth phase. As the product transcends into losing market share or market growth, this trait scans the external environments like the political changes introducing new regulations like those impacting generally accepted accounting principles (GAAP) or tax codes, economic indicators like the interest rates and impacting an organization's ability to raise the required capital to sustain development, social factors around the organization's willingness to embrace changes from outsourcing, technological constraints and opportunities in the horizon that the competitive forces are evaluating, and legal considerations surrounding privacy, financial accuracy, etc.

Practitioners can relate to the loss of market share from Netscape Navigator, Lotus Notes, and WordPerfect to Microsoft Internet Explorer, Microsoft Outlook, and Microsoft Office Suite respectively using its synch and stabilize approach to product development [50]. Simultaneously, Microsoft took the lead in changing their well established interface by introducing the ribbons in Office 2007 [51] while paving the foundation for a larger integration with their suites like Outlook, SharePoint, and other product lines. However, even such large organizations are not immune from competition; the Microsoft SQL Server faced competition from the open source packages like MySQL and Oracle's acquisitive strategies on the database front similar to Linux challenging the Windows operating systems. In the academic settings, the popular Learning Management Systems (LMS) like the Blackboard experience competition from the Moodle and the SaaS model based applications like from Edu20.

Flossing or exercising one day isn't going to guarantee health. It is an ongoing continual improvement process. Similarly, any successful product should constantly be groomed. Both the project management and agile development methodologies emphasize the Lessons Learned and retrospective sessions that must serve as a conduit for product development [52] to introduce new features and refinements in the releases. These lessons learned should be held frequently and address the efficiencies and economies of scale in getting more work done better, faster, and cheaper by assessing the onsite customer representation for field stories that should be translated into new requirements, target segments, and product associations using innovative channels, etc.

Specific objective metrics to look for measuring the groomability is particularly challenging because this trait focuses on unsaid intelligence gathering from the external field. However, a few thoughts to examine would be the following:

- The updated product risk register for uninsurable known external and technology business risks and the portion of contingency reserves allocated towards mitigating this risks

- The knowledge sharing activities to mitigate against the loss of intellectual human capital due to involuntary and voluntary attrition
- The burn rate of the action items identified from the product review sessions for information gathered through social media, customer surveys, focused interviews and trade shows that are followed through and closed on a timely fashion
- The rate of completing the user stories in the “Could” that are actually converted into “Should” and released according to the MoSCoW model [4, 18, 34]
- The reassessment of the “Won’t” user stories for their relevance to the product in comparison to the competition

3. CONCLUSIONS

There is a lot of discussion in the corporate environment on the responsibilities of product and marketing managers as the organizations embrace agile in their software development. Simultaneously many versions of agile framework evolve. There is little however in seeing the actual software product as a living entity and view the customers and the performing organization from the eyes of the software product. This exploratory paper is to personify that need giving life to the product using the PARAG Persona model with a set of attributes that any software product must always monitor. This approach would lead to evolving the roles and responsibilities for product owners and product managers, and those client facing roles that interface with the product development to view product development differently.

ACKNOWLEDGEMENTS

The author would like to thank all the customers, team members, and stakeholders involved in the various multi-project programs initiatives, individual projects and product developments the author had managed and the learners the author has trained in the corporate, professional, and academic settings. This paper is a summary of those lessons learned towards spreading the wealth of knowledge gained as a dedication to all these team members.

REFERENCES

- [1] Kuwashima, K. (2012). Product development research cycle: A historical review 1960s-1980s. *Annals of Business Administrative Science*, 11, 11-23
- [2] Anacona, D.G., & Caldwell, D.F. (1990). Beyond boundary spanning: Managing external dependence in product development teams. *Journal of High Technology Management Research*, 1, 119-135.
- [3] Dosi, G. (1988). Sources, procedures, and microeconomic effects of innovation. *Journal of Economic Literature*, 26, 1120-1171.
- [4] Cohn, M (2012). *Succeeding with Agile: Software development using Scrum*. Ann Arbor, MI: Addison Wesley.
- [5] Rajagopalan, S. (2014). Mapping of the product manager role to the product owner role. *Scrum Alliance*.
- [6] Brown, S. L. & Eisenhardt, K.M. (1995). Product development: Past research, present findings, and future directions. *Academy of Management Review*, 20(2), 343-378.
- [7] Lau, E.W.T. (2014). Empirical test of a new product evolutionary cycle. *Journal of Product & Brand Management*, 23(1), 43-54.
- [8] Gorchels L (2012). *The Product Manager's Handbook*. New York, NY: McGraw Hill.
- [9] Kotler, P. & Kotler, K. (2006). *Marketing Management*. 12th Edition. Upper Saddle River, NJ: Prentice Hall.
- [10] Patton, J. (2008). A conversation with Alan Cooper: The origin of interaction design. *IEEE Software*, 25(6), 15-17.
- [11] Pruitt, J.S. & Adin, T. (2006). *The Persona Lifecycle: Keeping people in mind throughout product design*. San Francisco, CA: Morgan Kaufmann Publishers.

- [12] Cleland-Huang, J. (2013). Meet Elaine: A Persona-Driven Approach to Exploring Architecturally Significant Requirements. *IEEE Software*, 30(4), 18-21.
- [13] Cooper, R.G. (2003). Profitable Product Innovation: The critical success factors. In. Larisa V.Shavina (Ed), *The Internatioanl Handbook on Innovation* (pp. 139-157). Kidlington, Oxford, United Kingdom: Elsevier Science, Ltd.
- [14] Crawford, C.M (1992). The hidden costs of accelerated product development. *Journal of Product Innovation Management*, 9(3), 188-199.
- [15] Di Nenedetto, C.A. (1999). Identifying the key success factors in new product launch. *Journal of Product Innovation Management*, 16(6), 530-544.
- [16] Cooper, R.G., Edgett, S.J., & Kleinschmidt, E.J. (2002). *Portfolio Management for New Products*. 2nd Edition. Reading, MA: Perseus Books.
- [17] Bergeron, B. & Blander, J. (2002). *Business Expectations*. Audio Tape Summary. John Wiley & Sons.
- [18] Rajagopalan, S. (2014, October). Agile metrics for organizational productivity. *Scrum Alliance*.
- [19] Schwaber, K. (2004). *Agile project management with scrum*. Redmond, Washington: Microsoft Press.
- [20] *A Guide to the Project Management Book of Knowledge* (2013). 5th Edition. Newtown Square, PA: Project Management Institute.
- [21] Yuce, O. & Arnold, J. J. (2014, July 28). Cost of Delay Workshop. *Agile 2014 Conference Proceedings*. Orlando, Florida.
- [22] Thompson, A.A., Strickland, A.J, & Gamble, J.E. (2005). *Crafting and Executing Strategy*. 14th Edition. Boston, MA: McGraw-Hill Irwin.
- [23] Williams, J. (2013). *Are you really in the service business?* Retrieved October 23, 2014, from <http://www.ignitiongroup.com/propulsion-blog-post/are-you-really-in-the-service-business/>
- [24] Rajagopalan, S. (2014, May). Evolution of account management in an agile environment. *Scrum Alliance*.
- [25] Rajagopalan, S. (2009, September 25). Influence of Emotional Intelligence on Leadership Skills of IS Project Managers with Virtual Teams. *Symposium Proceedings on Global Mindset Develop on Leadership & Management*, University of Riverside, California.
- [26] Prahalad, C.K., & Ramaswamy, V. (2004). *The Future of Competition: Co-Creating Unique Value with Customers*. Boston, MA: Harvard Business School.
- [27] Subramanian, N. & Chung, L. (2001, April). Metrics for software adaptability. *Proceedings of the Software Quality Management*, Loughborough, United Kingdom.
- [28] Chung, L. & Subramanian, N. (2001). Process-oriented metrics for software architecture adaptability. *Proceedings of Fifth IEEE International Symposium on Requirements Engineering*, 310-311.
- [29] Balaji, S. & Murugaiyan, M.S. (2012). Waterfall Vs V-Model Vs Agile: A comparative study on SDLC. *International Journal of Information Technology and Business Management*, 2(1), pp 26-29.
- [30] Rajagopalan, S. (2014, November). Review of the myths on original software development model. *International Journal of Software Engineering & Applications*, 5(16), 103-111.
- [31] Holzmann, G. (2013).Landing a Spacecraft on Mars. *IEEE Software*, 30(3), pp. 83-86.
- [32] Shull, F. (2014, January). Progression, regression, or stasis. *IEEE Software*, 31(1), 4-8.
- [33] Kaur, G. & Bahl, K. (2014, May). Software reliability, metrics, reliability improvement using agile process. *International Journal of Innovative Science, Engineering, & Technology*, 1(3), 143-17.
- [34] Cohn, M. (2004). *User stories applied for agile software development*. Boston, MA: Addison-Wesley.
- [35] Wasserman, A.I. (2011). Software Engineering Issues for Mobile Application Development. 2nd Annual Workshop on Software Engineering for Mobile Application Development, pp.1-4.
- [36] Kung, S.H. (2013). Seamless design approach for mobile applications. *International Journal of Advancements in Computing Technology*, 5(13), 532-540.
- [37] Flora, H.K., Wang, X., & Chande, S.W. (2014). An investigation into the mobile application development practices: Challenges and Best Practices, *International Journal of Modern Education and Computer Science*, 6, 1-9.
- [38] Juran, J.M. (1981). *Product Quality: A prescription for the West*. AMACOM.
- [39] Galdies, P. (2014). The insider threat to data assets. *Journal of Direct, Data and Digital Marketing Practice*, 15(3), 197-200.

- [40] Anand, I. M., & Anand, I. (n.d.) *Enhancing techniques and systems for logical games, activities and puzzles*. U.S. Patent Application 13/843,844.
- [41] Miller, R., & Collins, C. T. (2001). Acceptance testing. *Proc. XPUniverse*.
- [42] Davis, F.D., Walton, S.M., & Venkatesh, V. (2004). Toward preprototype user acceptance testing of new information systems: implications for software project management. *IEEE Transactions on Engineering Management*, 51(1), 31-46.
- [43] Meyer, B. (1998). The role of object oriented metrics. *Computer*, 31(11), 123-125.
- [44] Darejeh, A. & Singh, D. (2014). An investigation on Ribbon interface design guidelines for people with less computer literacy. *Computer Standards & Interfaces*, 36(5), 808-820.
- [45] Gwin, C.F. & Gwin, C.R. (2003, Spring). Product Attributes Model: A tool for evaluating brand positioning. *Journal of Marketing*, 30-42.
- [46] Ossadnik, W. & Kaspar, R. (2013). Evaluation of AHP software from a management accounting perspective. *Journal of Modelling in Management*, 8(3), 305 – 319.
- [47] Urban, G.L. & Hauser, J.R. (1993). *Design and Marketing of New Products*. 2nd Edition. Englewood Cliffs, NJ: Prentice Hall, 213-216.
- [48] Aaker, D. A. (1998). *Strategic Market Management*. 5th Edition, New York: John Wiley & Sons, Inc.
- [49] Thompson, A.A., Strickland, A.J., & Gamble, J.E. (2005). *Crafting and Executing Strategy*. 14th Edition. Boston, MA: McGraw-Hill Irwin.
- [50] Cusumano, M.A. & Selby, R.W. (1997). How Microsoft builds software. *Communications of the ACM*. 40(6), 53-61.
- [51] Dostal, M. (2010, November). User acceptance of the Microsoft ribbon user interface. *In Proceedings of the 9th WSEAS international conference on Data networks, communications, computers (pp. 143-149)*. World Scientific and Engineering Academy and Society (WSEAS).
- [52] Hlavacek, J., Maxwell, C., & Williams, J., Jr. (2009, July). Learn from new product failures. *Research Technology Management*, 31-39.

Authors

Dr. Rajagopalan graduated with a Bachelor's degree in Electronics and Communication Engineering from the University of Madras in India, a Master's (MS) degree in Computer Engineering from Wayne State University, Michigan, a business (MBA) degree in Management from Concordia University, Wisconsin, and a doctorate (PhD) degree in Organization and Management from Capella University, Minnesota.



Dr. Rajagopalan also holds many professional certifications in project management (PMP, PMI-ACP, PMI-SP, CSP, CSPO, CSD, CSM, ACC, IT Project+). With extensive software development and project management experience in many industries, he promotes the scholar-practitioner approach delivering information technology and management courses as adjunct faculty in several universities and community colleges in addition to delivering his own project and agile trainings.