

SIMULATION PROJECTS MANAGEMENT USING SCRUM

Eduardo J. Quaglia

Nokia Institute of Technology – INdT
Av. Torquato Tapajós, 7200 km 12
Manaus, AM 69093-415, BRAZIL

Claudia A. Tocantins

Atech Negócios em Tecnologia S/A
Rua do Rócio, 313, 2o andar
São Paulo, SP 04552-000, BRAZIL

ABSTRACT

The elaboration of simulation cases presents uncertainties, mainly before its modeling development starts. As the work advances, it's common to notice requirement changes on simulation projects due to the participants getting more and better understanding about the scope. The application of Agile methodologies, such as Scrum, can improve project management performance on simulation projects by the treatment of uncertainties. The performance improvement comes from the removal of no-value activities and treatment of: (1) activities plan; (2) clearer communication between customer and developers; (3) precise scope agreement; and (4) schedule definition. This paper proposes the usage of Scrum on simulation projects, based on Project Management Body Of Knowledge , Plan-Do-Check-Act cycle and Agile methodology. To validate the proposal, the paper presents a case of a simulation project in an electronics factory.

1 INTRODUCTION

Project management (PM) is a relevant aspect to be considered on simulation projects. Frequently, project customers, mainly the newcomers, have difficulty in understanding the simulation idea and think that the scope coverage is different than what it would be. Also, it is hard for customers to understand the complexity involved, consequently they barely have an idea of how long the simulation would take to be ready. On the other hand, considering a normal lack of specific business understanding, the simulation model developers (called modelers) may find it difficult to agree with the customer on the appropriate simulation scenarios and their expected results, making it hard to measure the needed workload. Timing is relevant since it is expected, in most of the cases, that the simulation schedule should be shorter than the actual scenario implementation. It is also requested, but not always possible, to get responses quickly. This way, choosing the right project management tools and techniques can improve a simulation project to: (1) define proper planning activities; (2) warrant clearer communication among stakeholders; (3) reach precise scope agreement and scheduling definition.

Using proper PM control, the project manager is able to avoid situations like “work that never ends” even though simulation projects frequently presents uncertainties and unpredictable behavior which can require more flexible control tools than the traditional PM techniques. For instance, when the customer evaluates the first scenarios results, he usually gets better understanding of the situation decreasing ambiguities, but resulting in more questions regarding the business case. Consequently the customer defines new scenarios that are sometimes quite different from the primary agreed scope (Schwaber 2004). The reader may come with the following questions: (1) Is the customer wrong not setting the definitive scope at the beginning of the project? (2) Is the developer wrong not helping the customer to define the scope before project starts? Probably, neither of that apply, since (a) more likely the customer is still learning what simulation is about, (b) the developer doesn't understand the process completely yet and (c) neither of them foresee the simulation results (otherwise they wouldn't need to simulate at first).

The most relevant reason for scope uncertainty might be that the simulation itself frequently indicates new ways of work when it reached its first results. Inspiration based on preliminary results is a peculiarity of simulation projects and it should be properly dealt with. So, the right question is how to deal with changes of scope/scenarios during the project? Traditional project management methodologies comprehend project change control process, but this paper suggests that a project management methodology based on Agile, particularly Scrum (Scrum Alliance 2011) - mostly applied on software development projects - fits simulation project characteristics better than traditional scope management techniques. Such methodology suggests iterative and incremental development, where the project scope is expected to change over time, giving preference to the most important needs/features.

The following sections present: Section 2 – Project Management main definitions, the PDCA cycle, its relations with simulation projects and the difficulties that can suggest the application of a specific methodology; Section 3 – the Agile Approach; Section 4 – the Scrum framework; Section 5 - a proposal to apply Scrum to simulation projects; Section 6 – a case of study, applying Scrum in a real simulation project; then conclusions and bibliography.

2 PROJECT MANAGEMENT

PMBOK (PMI 2008) is a worldwide recognized standard for project management profession, providing guidelines for managing individual projects, since it defines project management related concepts, project life cycle and related processes. PMBOK (PMI 2008) defines project as “a temporary endeavor undertaken to create a unique product, service, or result. The temporary nature of projects indicates a definite beginning and end.” This definition matches with simulation modeling projects which, to support a future actual implementation have: a) beginning and expected end; b) as final product a set of model scenarios for a specific situation (uniqueness), their result outputs and comparisons among the scenarios outputs in different perspectives.

Since we have a project, it's good practice to apply project management processes as recommended in PMBOK (PMI 2008), following the definition “Project management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.” Processes defined in the PMBOK are considered good practices on most projects most of the time.

The typical steps in a successful simulation modeling project are (Law 2007): a) problem assessment and requirements gathering; b) data collection, analysis and validation; c) model construction, verification and validation; d) experiments/scenarios design, conduction and analysis; e) documentation; and f) presentation of results.

The project activities can be summarized in the PDCA (Plan, Do, Check, Act) cycle (Figure 1). In a typical situation, the project manager is responsible for planning (P) the project. Once the plan is ready, the developers shall execute (D) it. During the execution the project manager does monitor (C) to see if execution is following the plan and performs the proper control (A) in case the project takes different directions. After the control step, the PDCA performs repeated cycles in itself until the product is ready for delivery and the project can be concluded.

Applying PDCA in simulation projects is quite similar to the typical steps above. It starts with defining the scope (object to be simulated, scenarios and objectives), and is followed by planning activities, allocating resources, running the activities, communicating project status with stakeholders, controlling occurrences and finally preparing, delivering and presenting the product/results. Simulation projects, however, present specific challenges for applying PDCA, which can make the development harder and sometimes less productive, specifically:

- Scope changes during project development. This can make the project less efficient, less agile.
- Preparation of an assumptions document (covering problem assessment and requirements, as stated by Law (2007)) – it can become a huge document, takes long preparation, and probably generates rework caused by changes. In this case, if the team keeps the document updated, the first version can be quite different from the last one.

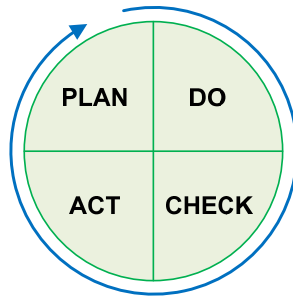


Figure 1: PDCA flow

- Difficult to perceive work evolution on long model preparation – consequently it may be hard to update customer with project status in the course of the project.
- For more complex models, these steps are even harder:
 - perform monitoring,
 - get appropriate control definition,
 - model validation (Law 2007)
- Communication between process engineers (who performs analysis and defines the model logics) and system engineers (who implements the model application) may present additional challenges. The simulation model is mainly developed by process engineers or someone similar, and not by software/computer skilled personnel. The process engineers may not necessarily be skilled or think about model development in a systematic way, which is more natural for software professionals.

Given that, is there a way of running PDCA in an agile perspective under simulation projects? How to anticipate the needed adaptations (changes) in the project scope and in its scenarios, that are so common under simulation? The Agile approach, described in the following section, may provide an answer.

3 AGILE

Ten years ago, a group of software development thinkers, sympathetic to the need for an alternative to documentation driven, heavyweight software development processes wrote what is known as The Agile Manifesto (Agile Manifesto 2001). They stated that “while there is value in processes and tools, documentation, contract and plans, we value individuals and interactions, working software, customer collaboration and responding to change more.” The Agile Manifesto summarizes a trend of best practices and methodologies that have grown in popularity under the software market since then (Griffiths 2004).

The idea of an Agile methodology was summarized by Highsmith (2009) “Be quick, but don’t hurry. In other words, do the right things, but learn how to do them quickly. Strip away the overhead, the non-value-adding activities. Create quality products and do it quickly. Agile development focuses on speed, mobility, and quality. To accomplish this, individuals and teams must be highly disciplined – but with self-discipline rather than imposed discipline.”

The Agile approach to project management began as a response to struggles that were being encountered in software development projects (Fewell et al. 2009). Agile project management is an interactive, incremental approach which recently gained force through the announcement of a PMI new credential (PMI 2011), since many of today’s projects operate in an environment of profound uncertainty and Agile can offer a framework to ensure acceptable project outcomes (Cottmeyer 2009).

The Agile methodologies focus on removing anything in the project that doesn’t create value to the product, produce it fast, and with the best quality. It is more related to adaptation of the work to the current necessity than to make detailed plans of what should be done. For instance, simulation has uncertainties, so the idea of an adaptation when planning the project is recommended instead planning each detail

till the end of the project. PMBOK (2004) defines Rolling Wave Planning as a form of progressive elaboration planning where the work to be accomplished in the near term is planned in detail and future work is planned at a higher level.

Some characteristics from Agile that can fit simulation environment are:

- Agile methodologies are people oriented with several iterations with customers.
- Agile aims to improve customer satisfaction through early and continuous delivery of valuable product. It usually comprises frequent deliveries along the project life.
- Agile intends to increase project productivity and reduce risks.
- People believe that there is no documentation in Agile, but Agile does include documentation.
- Agile can be used on small or large projects.
- In agile development, tests (Sawer and Brann 2009) and validations (Law 2007) are done frequently and integrated throughout the life cycle. As the project advances, less and less defects are found due to tests amount and frequency.
- Agile is especially useful when planners don't have a clear idea of the requirements beforehand and it is very effective on frequently changes of requirements.

Note: The above doesn't mean that these aspects don't happen on other approaches, but they are very natural on Agile.

Agile project management promises to deliver maximum value in uncertain problem domains where high quality and speed to market are your primary business drivers (Cottmeyer 2009). The two most popular agile methodologies, among several others, are: Extreme programming, also known as XP (Wells 2009), a software engineering methodology, and Scrum (Scrum Alliance 2011), with focus on software project management. It's valid to notice that XP and Scrum are not competitive methodologies and usually work well being applied together in the same project (FEROLEN 2011). This paper explores an approach using Scrum, which focuses on the management and control aspects of the project life cycle and seems to fit well on simulation projects.

4 SCRUM FRAMEWORK

This section presents a brief description of the main Scrum aspects and elements, including Sprint, regular meetings (planning, daily inspection), Backlog (Product and Sprint), stories, main roles (Product Owner, Scrum Master and Scrum Team), changing process and mission.

4.1 Scrum Definitions

PC Magazine (2011) presents the following definition: Scrum is an agile software development methodology developed by Ken Schwaber and Jeff Sutherland in the mid-1990s. Scrum is based on "Sprint" which is typically a 30-day period (Figure 2) for delivering a working part of the system. Each Sprint starts with a two to three-hour planning session that includes the customer (Product Owner), the facilitator (Scrum Master) and the cross-functional team (Scrum Team). The customer describes the highest priority in the "Product Backlog," and after the team agrees on how much of it to do (Sprint Backlog), the team is left alone to build it during the sprint duration. To keep the team synchronized, there is a 15-minute meeting every working day called Daily Inspection. At the end of the Sprint, the results are delivered (Figure 3), reviewed, and the next Sprint is started.

4.2 Backlog

The "Product Backlog" is a set of functionalities (called "Stories" on Scrum) expected to be implemented until the end of the project and the "Sprint Backlog" is a subset of the Product Backlog (Figure 4) expected to be implemented in the current sprint. The backlog stories are sorted in the Product Backlog according to its importance to the customer and the business value, and the Sprint Backlog shall contain the current most important stories which are feasible to be developed in its sprint time frame.

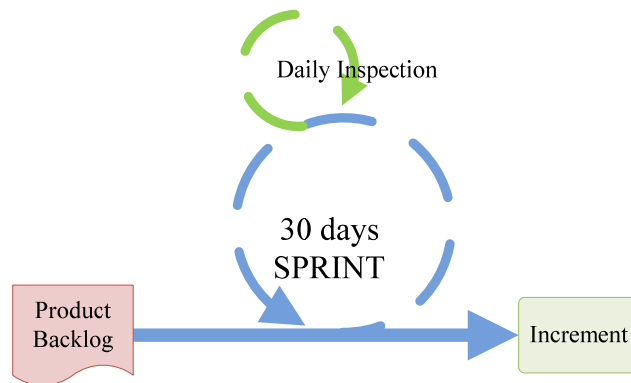


Figure 2: Scrum - 30 days sprint

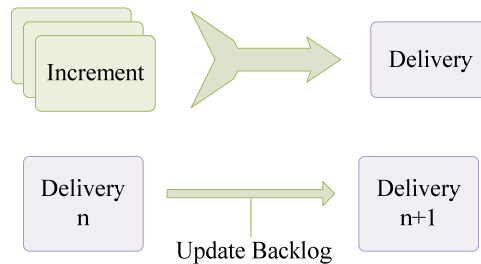


Figure 3: Scrum incremental deliveries

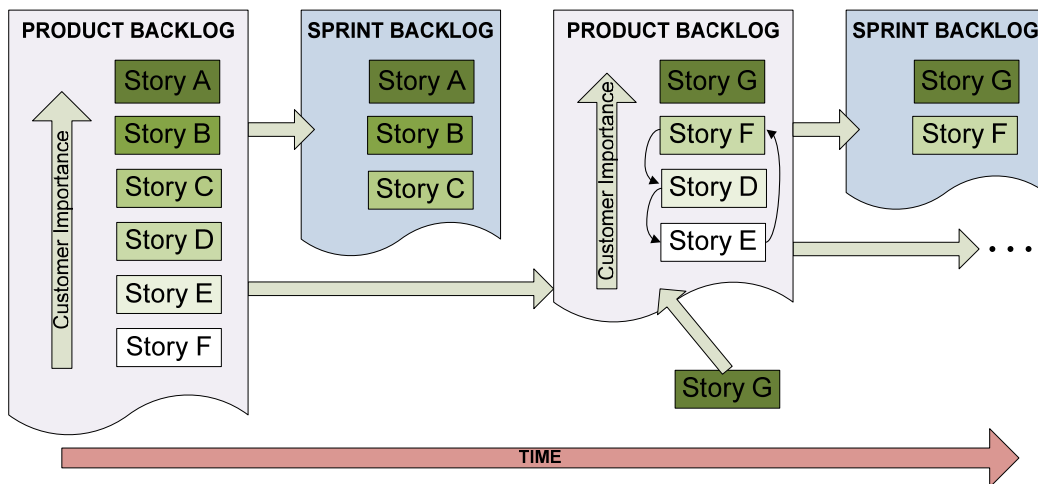


Figure 4: Product Backlog & Sprint Backlog.

At the end of a Sprint and before selection of the next Sprint Backlog stories, the Product Backlog shall be revised by the Product Owner in order to add and/or remove stories and sort the remaining ones again by priority (Figure 4). The priorities are primarily based on the project mission, but also on market behavior, new insights gotten by the customer in the last period and his/her perception after running the already implemented part of the product.

Sprints are kept short enough that the stakeholders don't lose interest in the project before the Sprints are completed. And stakeholders can see that they have an opportunity to redirect the project at the start of every Sprint to optimize the value they derive from the project. At the end of every Sprint, stakeholders see new functionality (Schwaber 2004).

Notice that it's not mandatory that the whole set of Product Backlog stories be implemented until the end of project. With the project and sprints progress, the remaining stories in the Product Backlog are always the less important ones, the customer already has a partial useful product with the most important features, so he/she has the chance to stop the project if he/she understands that the remaining business case is not big enough to continue or the current delivered product is functional enough. The secret is to follow the rule of "do not stop or change a sprint during its execution." All the changes must be set in between each two sprints.

4.3 Mission

About the Project Mission, Appelo (2009) writes "A mission statement tells you the fundamental purpose of the project." And he says that "this statement shall be a) concise, b) plain, c) inspiring, d) useful, and e) memorable." This is strongly relevant to Scrum since the team has to make decisions daily, inspire itself and the others and produce something useful in a plain and concise way in order to reach the milestones. A clear mission in mind makes it easier to achieve project success.

4.4 Roles

By Scrum definition, there are three roles in the Scrum framework: Product Owner, Scrum Master and the Scrum Team.

- **Product Owner (PO)** – represents the customer in the project. He is responsible for setting the project mission, writing the Product Backlog stories and receives/validates the product released at each Sprint. He is expected to answer questions from the team related to scope.
- **Scrum Master (SM)** – works as a facilitator of the process, resolving project impediments, in order to guarantee that the team doesn't lose its focus on producing valued outputs and facilitating the communication between the PO and the Team. He is not someone expected to have authority over the Team or PO.
- **Scrum Team (ST)** – is the group that selects the Sprint Backlog stories (based on the team possibilities) and develops, validates and delivers the stories functionalities. The team doesn't have a hierarchical organization, it manages itself with self-commitment to deliver the whole set of stories in the Sprint Backlog on time. In general, the ST people are allocated to a sprint only if they are useful to that specific sprint. This way, not required resource may not work during the whole project as ST member, even though practice says it is more productive if the team remains the same since the Sprint estimation gets better with the project evolution.

5 SIMULATION UNDER SCRUM & PDCA

This article's core idea is to understand how to apply Scrum in the perspective of PDCA and Simulation Modeling environment. The following topics address each main Scrum activity in each of the Simulation/PDCA project steps presented in section 2.

5.1 Defining the Scope

- Preparation of the Product Backlog. In other words, define the mission (Appelo 2009) of the project and describe what has to be accomplished in stories to fulfill the Product Backlog. This looks like the main structure of the assumptions document (Law 2007).
- Each simulation scenario would be built by a subset of Product Backlog stories.

- Under Scrum, the stories have to be prioritized in some way pertinent to customer needs - and the development has to follow this prioritization.

5.2 Planning Activities

- Scrum Team chooses the Sprint Backlog stories to be implemented on the following Sprint.
- Scrum Team defines the tasks to accomplish the Sprint Backlog stories. Notice that it shall have no task definition for stories out of the Sprint Backlog. Under simulation the tasks for a hypothetical story “simulate the module A in the scenario 1” could be: a) determine the logic flow of the module; b) collect time data of one point in the model (should be one task for each point); c) define the statistical distribution for the point for which data was collected (one task for each point); d) run experiment (one task for each specified experiment); etc.
- Scrum Master supports the planning meetings/activities in order to eliminate impediments and facilitate communication among Product Owner and Scrum Team members.

5.3 Allocation of Resources

- Team members are assigned exclusively to the Sprint and would not lose focus (resource allocation is 100% to project during a running Sprint) until the end of the Sprint.
- The members of the team define, daily, who is going to do what, considering the tasks they had defined for the Sprint backlog.
- Depending on the Sprint Backlog, the team members could be, for instance, statistical analyst(s), simulation model developer(s), software developer(s), technical interviewer(s), documenter and designer.
- The Product Owner and Scrum Master are allocated to the project, consequently to all the Sprints.

5.4 Running the Activities

- The team members select daily and work on the stories tasks.
- Each team member briefly explains daily what he/she has done in the last 24 hours, what he/she intends to do in the next 24 hours and what are the impediments to execute his/her tasks.
- Under Scrum the team is supposed to be “isolated” from the external world in order to get focus on the stories’ work. It seems hard under a simulation modeling project when the modeler usually needs to frequently meet with technical customers in order to get more detailed information, perform measurements or clarify aspects of the process. Even then, it is suggested that the team isolate itself, when possible.

5.5 Communicating Project Status to Customer

- Formally, at the end of each Sprint the team meets the Product Owner to provide the stories fully developed and to explain the reasons part of the Sprint scope (stories) were eventually not finalized.
- Informally, the Product Owner may have access to the team at any time during the Sprint execution.
- The Product Owner updates the stakeholders with status in a suitable way.

5.6 Controlling Occurrences

- The Scrum Master works to facilitate the communication among the team members and to solve any development impediments.
- At the end of each Sprint the Product Owner reevaluates the Product Backlog, creates/removes/rewrites remaining stories and updates the stories prioritization in order to guarantee

that the team will work on the current most important activities and scenarios. That is the case, for instance, when the customer sees the results of one scenario and changes his/her mind, creating/removing/redefining the remaining scenarios.

5.7 Preparing, Delivering and Presenting the Product/Results

- At the end of each Sprint the team is supposed to have an increment of the solution/analysis/documentation which should be delivered to the Product Owner and consequently to the customer.
- The results shall be incremented Sprint by Sprint. In the same way, the assumptions document should be updated during the Sprint. At the end of the project the features indicated on the assumptions document should be completely implemented and validated. This way, it is expected to minimize rework on the assumptions document and it provides agility for the artifact completion.

6 CASE OF STUDY

This section presents the application of the approach proposed above on an industrial simulation project for a factory of electronic devices. For this paper's purpose, the project is called MDL.

6.1 Environment

The MDL project consisted of simulation of distinct pieces of the process in a factory of electronic devices. Each piece to be simulated becomes a subproject. The simulation subprojects were related to production engineering including aspects of production process improvement, material flow and resources flow analysis inside the factory. A consultant company external to the factory was contracted to run the MDL Project.

The consultant company had extended experience on using Scrum for software development, including in projects for the same client. However, it had never executed a simulation modeling project using an agile methodology. Since the results of applying agile on software development were good, the team assigned to MDL had the idea of expanding the application to a simulation modeling project, believing that it could improve performance and customer satisfaction.

6.2 The team

A team of 7 people was allocated to run the subprojects (Product Owner, Scrum Master and 5 modelers). From the customer side, there were 2 to 4 people involved in each subproject as managers and technical specialists, providing technical answers and directions to the team and giving needed support. Half of the development team had a few years of experience running simulation projects in a traditional way but only theoretical knowledge about agile methodologies. The other half had no previous experience with simulation but had with software development using Scrum. They were suitable to work on the process analysis and simulation modeling since they also knew the factory (building and processes) of the MDL project pretty well. All of them took similar roles in the project such as process analysis, acquiring data, model development, running experiments and reporting.

For the part of the team with agile experience, it was quite easy and natural to follow Scrum, defining the scope in backlog stories, breaking each in tasks and communicating with the rest of the group about the evolution of the work. The simulation itself was not a concern.

For the others with no previous Scrum experience, changing their minds to agile was not that easy. At the beginning, Scrum seemed more a disturbance than a help for the simulation work that they used to perform well. As the project advanced, the advantages and benefits of following the Agile methodology became clear to everyone.

6.3 Observations

Finding of this Simulation/Scrum experience:

- It was realized that the participation of customer on modeling stories is bigger than on software production. So, it's harder not to have changes on stories during the Sprint. An optional work-around, which seems to be closer to Scrum framework, would be to break the stories in smaller ones, reducing the scope of each and reducing the Sprints' durations. The project started with Sprints of 2 weeks but changed to 1 week Sprint duration. Usually software projects run Sprints of 2 to 4 weeks.
- Another option for managing changes during Sprint is to assure that the developer can discern when the customer asks for improvement during the sprint. If the developer find out that the change could be accomplished until the end of the sprint, not impacting the other tasks, he/she would add the needed tasks to the story and proceed with the development; otherwise he should settle with customer to leave it to a new story in another Sprint. On the Agile point of view, this option is proactive to customer needs, but it affects considerably the Scrum control, and the developer usually feels that the Scrum method has been impaired.
- The tool used for simulation, Arena (Rockwell Automation 2011), is monolithic (full model in a unique file) and not objected oriented and doesn't allow many people working at the same model. In this aspect, XP methodology (Wells 2009) would be an option to improve performance, since it relies on pair programming, as one of its basic heuristics. The application of XP in this case would fit on Scrum with no added implications.
- A proper simulation tool should be considered on a future project. Simio (SIMIO 2011) framework is under evaluation to replace Arena. It is object oriented and easier to distribute the work among the team members.

7 CONCLUSIONS

The application of an agile methodology, like Scrum, to simulation projects could improve project effectiveness since this approach works well for scope change control which happens frequently on simulation. Previous experience of Scrum on software development (in its more traditional usage) makes it easier to apply it to simulation modeling.

An object oriented tool, instead of a monolithic one, would improve productivity, since it would provide more flexibility on the task distribution among the Scrum team members, a good subject for another article.

The agile approach is gaining popularity, probably due to a larger demand on innovation projects which are, by definition, filled with uncertainties and it will overcome the limits of the software projects area. People oriented projects are also good candidates to benefit from agile.

ACKNOWLEDGMENTS

The authors would like to thank the MDL participants.

REFERENCES

- Agile Manifesto 2001. "Manifesto for Agile Software Development" Accessed March 31, 2011. <http://agilemanifesto.org/x>
- Appelo, J. 2009. "What Is the Mission of Your Project?" Accessed March 25, 2011. <http://www.projectsmart.co.uk/what-is-the-mission-of-your-project.html>.
- Rockwell Automation. 2011. "Arena Simulation Software." Accessed March 24. <http://www.arenasimulation.com/>.

- Cottmeyer, M. 2009. "Agile PMP: Managing Software Projects in the Face of Uncertainty." In *2009 PMI Global Congress – North America Proceedings*, Orlando, Florida. Newtown Square, Pa: Project Management Institute.
- FEROLEN. 2011. "XP Vs Scrum". Accessed June 29 <http://www.ferolen.com/blog/xp-vs-scrum/>.
- Fewell, J., M. Jack, D. Prior, P. Rosado, and B. Tarne. 2009. "Challenges in Implementing Agile Project Management." In *2009 PMI Global Congress - EMEA Proceedings – Amsterdam, Netherlands*. Newtown Square, Pa: Project Management Institute.
- Griffiths, M. 2004. "Utilizing Agile principles alongside A Guide to the Project Management Body of Knowledge (PMBOK Guide) for better project execution and control in software development projects." In *2004 PMI Global Congress – North America Proceedings – Anaheim, California*. Newtown Square, Pa: Project Management Institute.
- Highsmith, J. 2009. *Agile Project Management: Creating Innovative Products*. 2nd Edition. Addison-Wesley Professional.
- Law, A. M. 2007. *Simulation Modeling and Analysis*. 4th edition. McGraw-Hill.
- PC Magazine. 2011. "Definition of: Scrum, from Encyclopedia." PCMAG.COM. Accessed March 25. http://www.pcmag.com/encyclopedia_term/0,2542,t=Scrum&i=50946,00.asp
- PMI. 2008. *A Guide to the Project Management Body of Knowledge (PMBOK)*. 4th edition. Newtown Square, PA: Project Management Institute, Inc.
- PMI. 2011. "PMI Agile Certification" Accessed March 31, 2011. <http://www.pmi.org/Agile/Agile-Certification-Eligibility-Requirements.aspx>.
- Sawyer, J., and D. Brann. 2009. "How to Test Your Models More Effectively: Applying Agile and Automated Techniques to Simulation Testing." In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 968-978. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Schwaber, K. 2004. *Agile Project Management With Scrum*. Redmond, WA: Microsoft Press.
- Scrum Alliance. 2011. "What is Scrum?" Scrum Alliance. Accessed March 24. <http://www.scrumalliance.org/>.
- SIMIO. 2011. "Simio Simulation Software" Accessed March 24. <http://www.simio.com/index.html>.
- Wells, D. 2009. "Extreme Programming: A gentle introduction." Accessed March 24, 2011. <http://www.extremeprogramming.org/>.

AUTHOR BIOGRAPHIES

EDUARDO J. QUAGLIA is a Project Manager at the Process & Simulation group of Nokia Institute of Technology (INdT). He holds a B.S. in Applied Mathematics, a M.E. in Electrical Engineering with focus on Operational Research, and a Post-Graduation in Innovation Management, all from University of Campinas – UNICAMP, Campinas, SP, Brazil. His research interest is on project management for modeling and simulation, and innovation management. His email address is eduardo.quaglia@indt.org.br.

CLAUDIA A. TOCANTINS is a PMP, certified by PMI, currently volunteering as Director of Certification for PMI Manaus Chapter. She has over 18 years of experience on system and software analysis and design in the following areas: wide-area surveillance, defense, environmental management, geo-processing and spatial database. She holds a MSc in Computer Science and a Computer Engineer degree both from PUC-Rio, Rio de Janeiro, Brazil. Her email address is claudia@atech.com.br.