Managing Intellectual Capital and Innovation for Sustainable and Inclusive Society
27–29 May 2015
Bari • Italy
Management, Knowledge and Learning Joint International Conference 2015
Technology, Innovation and Industrial Management

RISK MANAGEMENT FOR SOFTWARE PROJECTS IN AN AGILE ENVIRONMENT – ADOPTING KEY LESSONS FROM THE AUTOMOTIVE INDUSTRY

Oana Iamandi
Techincal University of Cluj-Napoca, Romania
oana.iamandi@muri.utcluj.ro

Marius Dan
Technical University of Cluj-Napoca, Romania
marius.dan@calitop.ro;

Sorin Popescu
Technical University of Cluj-Napoca, Romania
posorin@gmail.com

Abstract:
The present paper seeks to improve the risk management practices undertaken in software development projects in agile environments. Software development projects, clearly having benefited from a steep incensement in both number and importance, is reportedly lagging behind in key project management areas, one of which being risk management. By providing key insights from one of the most successful and most rigorously structured industries – the automotive manufacturing – we aim to bring new value to the risk management practices in the form of a set of tools and techniques. The paper is strongly based on both a thorough relevant scientific literature investigation and the personal experience of the authors in both software and web development project management and quality and risk management systems in the automotive industry. In developing the theoretical part of the paper, we employed tools from and techniques from the fields of information structuring, quality management and engineering and competitive engineering. The scope of the paper is restricted to a theoretical framework, by proposing a context resulted from the over-positioning of APQP as a Stage-Gate model and the structure of the software development lifecycle adapted for to an agile environment. The application of the risk management process and the associated proposed tools and techniques are in strict adherence with the previously mentioned context.

Keywords: risk management, project management, agile environment, stage-gate model, APQP model, information technology

1. INTRODUCTION

As much of the literature on Agile project management and agile software development agrees, the agile environment, although until recently very utilized and proclaimed more successful than other methodologies, still lacks in rigor and structure. Until recently, the performance requirements of software development projects were quite low and these conditions spawned a plethora of companies and individuals dealing with just that to their best of their competence. In the current economic context, such approaches leave much to be desired and, as a result, the practices have very quickly shifted towards better performance of software projects and higher delivered value with the same costs. The current high failure rate among software development projects demonstrates that this is simply not enough. As a response, practitioners and academics are striving to create and adapt the best processes, tools, techniques and best practices to better accommodate the current peak in demand for a new type of software development projects and their management.

As a result, this paper is trying to respond to the need of structure in the area of risk management related to software development projects. By applying both a thorough research process, a critical approach and the experience of the authors in software development project management and the automotive industry, we attempted to apply the lessons learned from one of the most standardized and thorough industries – automotive manufacturing – into a very dynamic and emergent by comparison software development industry.

2. LITERATURE REVIEW

The focus of the literature review is twofold: (1) to identify the practices and applications of risk management in both stage-gate models and agile environments as well as to spot the deficiencies and shortcomings of each of the two in relation to risk management, and (2) to identify previous work done on adapting different tools and techniques from one area of application to another and possibly “cocktail” models.

2.1. Stage-Gate model

The Stage-Gate model was developed by dr. Robert Cooper and was initially used for New Product Development. The general Stage-Gate model has long been used as the preferred approach in developing new products. It consists of a series of “gates”, placed at the end of each stage of the process that act as both decision enabling and as corrective mechanisms. Each gate requires a set of deliverables, which in turn have been previously designed in the project plan. Each stage gate also involves customer or user feedback so as to allow the passing to the next stage if validation was received (Cooper, 1994).

Being a generic process, the Stage-Gate framework benefits from many adaptations in various industries and fields. The Advanced Product Quality Planning – APQP is a very good adaptation in the automotive field and displays all the components of a mature methodology (Chrysler Corporation, Ford Motor Company and General motors Company, 2008): structure, process, tools and techniques, industry reach and a number of significant case studies form both practitioners and academia.

2.2. Agile environments

Agile processes have been in place for a number of years, even if not by name. The flexible manufacturing paradigm is an early manifestation of this “school of thought”. With the appearance of the Agile Manifesto (Beedle, et al., 2001), agile received its name and has taken up ever since more and more adopters. (Conboy, 2009) defines software development agility as the continued readiness “to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment.”

The current agile practices have little theoretical foundation, with just a few frameworks defined, and lack in “establishing theoretical underpinnings, when investigating agile development and its various practices” (Dingsøyr, Nerur, Balijepallyd, & Brede Moea, 2012). The observation is valid also in the area of risk management where, although we could identify some implicit practices (Noor & Khan,

2014) – especially in the scrum methodology (Alharbi & Qureshi, 2014) – there is little attention paid and even lesser structure and dedicated tools.

Risk management in agile environments opens a new perspective on agility in organizations. It proposes a new, more mature and comprehensive agile model where agile models and frameworks are not constricted to the people and team size (Kettunen, 2009) – it integrates agility at the enterprise level and proposes that the deployment be done downwards instead of upwards.

The future of Agile, as seen from the literature review is to incorporate new structured tools and techniques and even to allow hybrid models, customized towards industrial specific practice and even to company specificity. As (Boehm & Turner, 2004) concluded, the future of projects involves both high agility and great discipline.

2.3. Models and related literature

In (Kettunen, 2009) the author creates a link between agile manufacturing and the new product development process. He states "some of the roots and basic ideas of agile software development appear to have much in common with the origins of AM (n.a. agile manufacturing) (NPD)". The same interesting parallel between NPD and AM also appears in (Buyukozkan & Baykasoglu, 2004), where AM is stated to have "useful enabling technologies and physical tools which may support the efficient product development". However apparent it may seem that there are great benefits from adopting agile principles in manufacturing, we must retain in mind that agile manufacturing benefits from the historical development of manufacturing management systems, providing a sound basis for structured and comprehensive processes and tools in order to deliver best results. The agile applications in the software development are far newer and do not benefit from an established context.

In (Cooper), the author clearly demonstrates the compatibility between the two. He argues that, although successful in delivering fully working products or increments by the end of every sprint, agile is not appropriate for all the project lifecycle. This is why it is mostly used by technical departments and staff and cannot reach the whole organization, thus limiting the scope of the project.

3. RESEARCH METHODOLOGY

The research starts from the premises proposed by (Kettunen, 2009) which concludes, "there is clearly a need to expand the view in software engineering, adopting applicable key learnings from other disciplines". Moreover, in the case of agile practices, there is a clear need for more structured and established risk management processes and practices. As a result, this paper attempts to adapt the lessons learned form an established and well-standardized industry, the automotive sector, and bring them into an agile context.

3.1. Research questions and methodology

In order to achieve its objective, we posed a series of research questions:

- Q1: Are Stage-gate type models and Agile models incompatible?

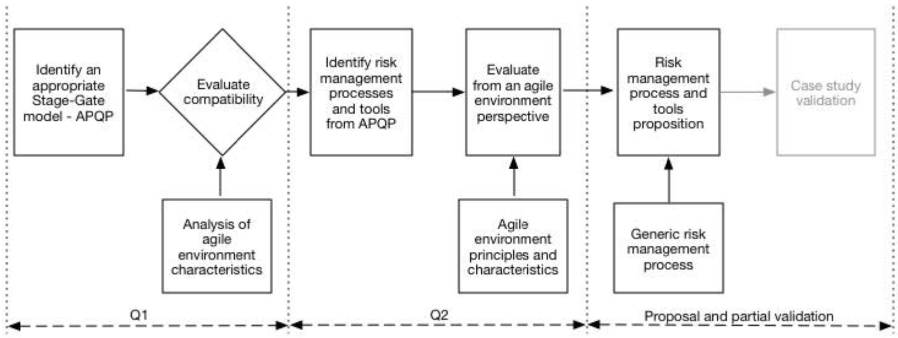
The objective of this question is to identify whether a parallel implementation or even a mixed model of agile and Stage-Gate is possible. The question poses great importance, as most of the well-established management and risk management models are a practical application of the Stage-gate framework. When attempting to reconcile two types of models or to adapt parts of one for another it is essential that we previously validate whether the two belong or not to two mutually exclusive philosophies.

- Q2: If Q1 is infirmed, what can a Stage-Gate model bring towards the improvement of the Agile methods and practices?

The second question represents the fundamental premise of the research. It attempts to identify the elements which (1) have a risk management dimension and (2) are compatible and transmissible towards the agile environments. Answering the second question creates the basis for developing an

adapted risk management framework, a risk management tools and techniques roadmap, specifically adapted for agile software development projects.
The structure of the paper follows the methodology proposed in Figure 1.

Figure 1: Research methodology



For the Stage-Gate model, we chose the Advanced Product Quality Planning – APQP model, widely utilized in the automotive industry. The choice was motivated by two factors: (1) the automotive industry is one of the most prolific and the most successful in developing new products. It is also one of the highly standardized, regularized and controlled industries, thus benefiting from designated frameworks and tools. Lessons learned from the automotive environment have been repeatedly proven valuable in other environments such as other manufacturing areas, services and others. (2) the authors have recognized experience in the implementation and use of APQP model.

For evaluating the risk management tools and techniques we used the QFD (Quality Function Deployment) tool to help us deploy the tools through the agile principles and characteristics. QFD is a competitive engineering structured instrument that is used primarily in quality management and quality planning both for processes and products. Its aim is to identify the primary characteristics that meet the customer requirements and to deploy them into the product or process. We used a simplified QFD template called the House of Quality (HOQ) as follows:

- The principles and characteristics of agile environments – scrum in particular – were identified and previously assigned an importance factor (in relation to their impact on risk management) using Analytical Hierarchy Process. They further become inputs – as requirements – in the QFD HOQ matrix;
- The risk management tools and techniques identified from the APQP are the second input for the QFD HOQ matrix, which will be deployed through the agile requirements and the most relevant will become the input for the customized risk management process adapted for software development agile environments;
- The roof of the QFD HOQ matrix was used to eliminate any redundancies that might occur from applying two similar tools and techniques, through solving the identified conflicts.

4. RESULTS AND DISCUSSIONS

4.1. Demonstrating the compatibility between stage-gate and agile

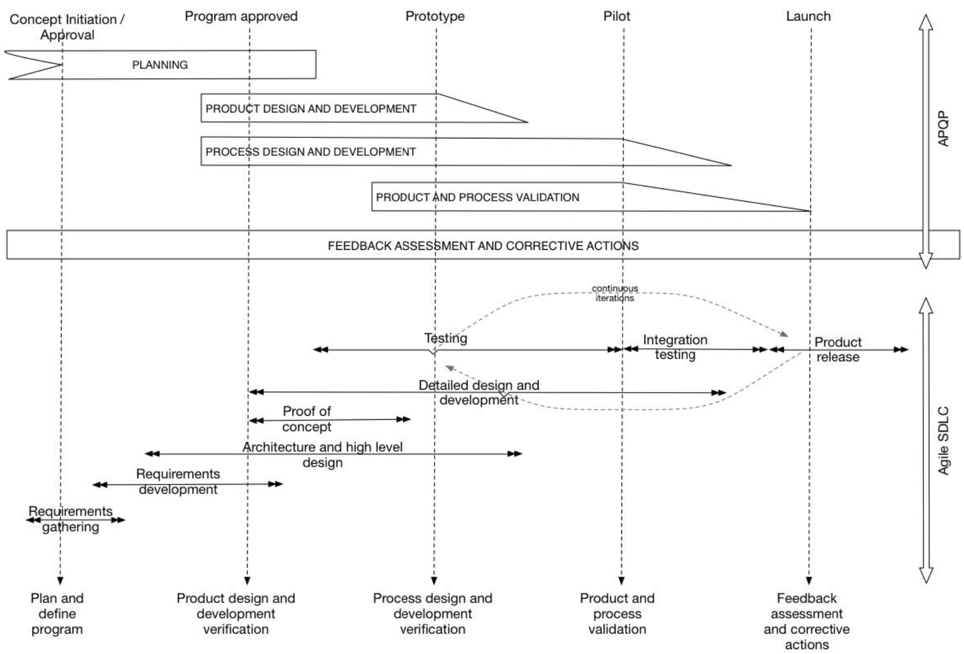
In the first step of the paper we deal with Q1 and validating whether the Stage-Gate model and agile are compatible and complementary or not. The second part deals with identifying and adapting risk management tools and techniques. In order to validate Q1, we chose the most widespread and adopted Agile framework that is currently in practice and high demand in terms of application and know-how: Scrum.

For the purpose of validating the first research question we employed two methods. The first is a review of the relevant literature. (Dyba & Dingsøyr, 2008) found through a literature study that “agile methods give the stage-gate model powerful tools for micro-planning, day-to-day work control, and reporting on progress”. As well, “the stage-gate model provided the agile methods with a means to coordinate with other development teams and to communicate with marketing and senior management”. They therefore conclude that it is highly feasible and possibly even beneficial to integrate the two schools of thought.

Probably the most relevant view regarding the compatibility between the Stage-Gate model and Agile comes from the developer of the model, Robert Cooper. He states that, in response to the criticism that this model has faced during the years (since its development in the 1980s) regarding it being “too linear, too rigid, and too planned to handle more innovative or dynamic projects” (Cooper, 2014), the model has greatly evolved through the years (Cooper, 1994), (Cooper, Edgett, & Kleinschmidt, 2002) and (Cooper, 2011). The findings in (Cooper, 2014) show that many leading companies have integrated key elements of the Agile Manifesto into their own adaptation of the Stage-Gate model – when discussing about manufacturing processes. We can therefore conclude that while agile establishes a context, a set of guiding rules and the principles of conducting a project, it is the stage-gate model that gives the structure and can bring the tools and techniques to effectively manage the risks and the processes as a whole. The result of the literature analysis is that there is no impediment in trying to reconcile the two models, and in some cases it is even recommended (Kettunen, 2009).

For the second method we attempted an over-positioning of the process model of APQP and SDLC (software development lifecycle) – following Agile Scrum principles –, which is graphically depicted in Figure 2. As it can be observed, not only are the two not mutually exclusive, but, when aligned correctly (as the authors attempted in Figure 2) the SDLC can greatly benefit from applying a Stage-Gate model in general and the APQP model in particular. In the agile environment context, the stage-gate framework should act as a navigation guide throughout the product and project lifecycle, to aid in defining different milestones and to offer the appropriate inputs and outputs for the risk management process.

Figure 2: Over-positioning of APQP and SDLC



4.2. Identifying the improvement points for agile processes derived from APQP

In order to answer the second research question, we must first identify the risk management processes and tools employed in the APQP model. As a Stage-Gate model, APQP uses milestones as gates in order to verify the results of the stage that has just ended and to provide a decision opportunity. In itself the process is a very good risk management application, enabling GO / NO GO decisions at the end of every major stage and delivering stag outputs that are relevant for adjusting the course of action. The risk management tools and methods employed in APQP are detailed in the table below:

Table 1: Risk management tools and techniques in APQP (Chrysler Corporation, Ford Motor Company and General motors Company, 2008)

	Inputs	Outputs
Concept initiation and approval	• Voice of the customer • Data and information from the client • Hypotheses relating to the product and the process	• Project objectives • Quality objectives • Process flow diagram • Quality and risk assurance plan • Product generic specifications
Program approved	• Project objectives • Quality objectives • Process flow diagram • Quality and risk assurance plan • Product specifications	• Risk and quality system analysis • Design specifications • Equipment, tools and verification instruments requirements • Project analysis and verification • DFMEA – design failure modes and effects analysis • Prototype
Prototype	• Risk and quality system analysis • Design specifications • Equipment, tools and verification instruments requirements • Project analysis and verification • DFMEA – design failure modes and effects analysis • Prototype	• Product characteristics matrix • Work instructions • PFMEA – process failure modes and effects analysis • Process capability study plan
Pilot	• Product characteristics matrix • Work instructions • PFMEA • Process capability study plan	• Measurement system evaluation • Process capability – preliminary study • Control plan for product launch
Launch	• Measurement system evaluation • Process capability – preliminary study • Control plan for product launch	

In order to established the requirements for the tools and techniques which will be adapted to the agile environment we developed a list of hierarchized characteristics of agile environments – with a focus on the Scrum methodology – which were defined starting from the principles of agile defined in the Agile Manifesto (Beedle, et al., 2001) and the work presented in (Miller, 2001).

Table 2: Agile environments characteristics hierarchized

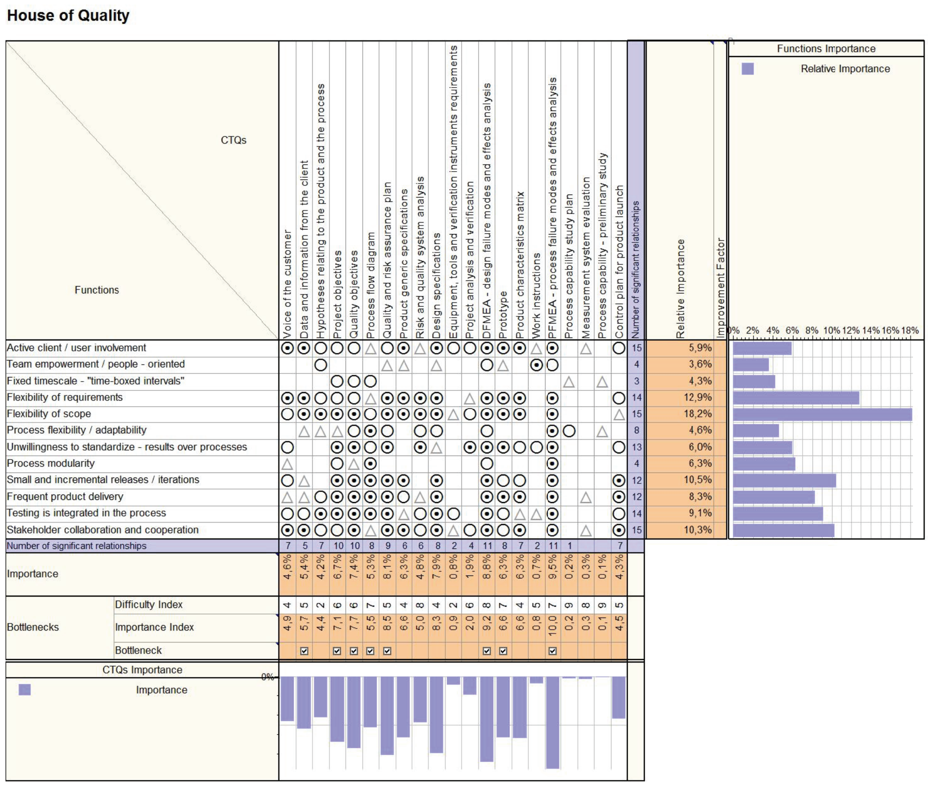
Agile characteristics (requirements)	Ranking	Importance in group [%]
Active client / user involvement	8	5.9%
Team empowerment / people – oriented	12	3.8%
Fixed timescale – "time-boxed intervals"	11	4.4%
Flexibility of requirements	2	13.2%
Flexibility of scope	1	18.8%
Process flexibility / adaptability	10	4.8%
Unwillingness to standardize – results over processes	9	5.7%
Process modularity	7	6.1%
Small and incremental releases / iterations	3	10.9%
Frequent product delivery	5	9.1%
Testing is integrated in the process	4	9.2%
Stakeholder collaboration and cooperation	6	8.2%

The ranking was achieved by applying an AHP matrix using the Qualica QFD software tool and it represents the criticality of the characteristic in the risk management process. This means that the higher the rank, the higher importance of the characteristics in the risk management process, As stated in the research methodology, the QFD HOQ technique was used in order to deploy the previously identified risk management tools and techniques through the agile requirements. The results are presented in Figure 3.

When discussing the results of the risk tools and instruments deployments, result a number of eleven tools considered relevant in the context of agile environments and processes. In order of their importance these are: PFMEA (9,5%), DFMEA (8,8%), Quality and risk assurance plan (8.1%), Design specifications (7,9%), Quality objectives (7,4%), Project objectives (6,7%), Product generic specifications (6,3%), Prototype (6,3%), Product characterization matrix (6,3%), Data and information from the client (5,4%), Process flow diagram (5,3%). Considering that leanness of the process is a high consideration in developing the tools and techniques map, we will consider the very similar Product generic specifications, Design specifications and Product characterization matrix into one tool: Product specifications with an averaged importance factor of (6.8%).

Also a number of bottlenecks emerged, because of the difficulty of the implementation and support of the different tools and instruments (in order of their importance): PFMEA (9,5%), DFMEA (8,8%), Quality and risk assurance plan (8.1%), Quality objectives (7,4%), Project objectives (6,7%), Prototype (6,3%), Data and information from the client (5,4%), Process flow diagram (5,3%). The bottlenecks function as indicators of the most important tools and techniques that have emerged from the deployment analysis. We have used the bottleneck analysis to identify the critical to risk management tools to be implemented and, as a consequence, adapted to the agile environment.

Figure 3: QFD House of Quality for risk tools and instruments deployment

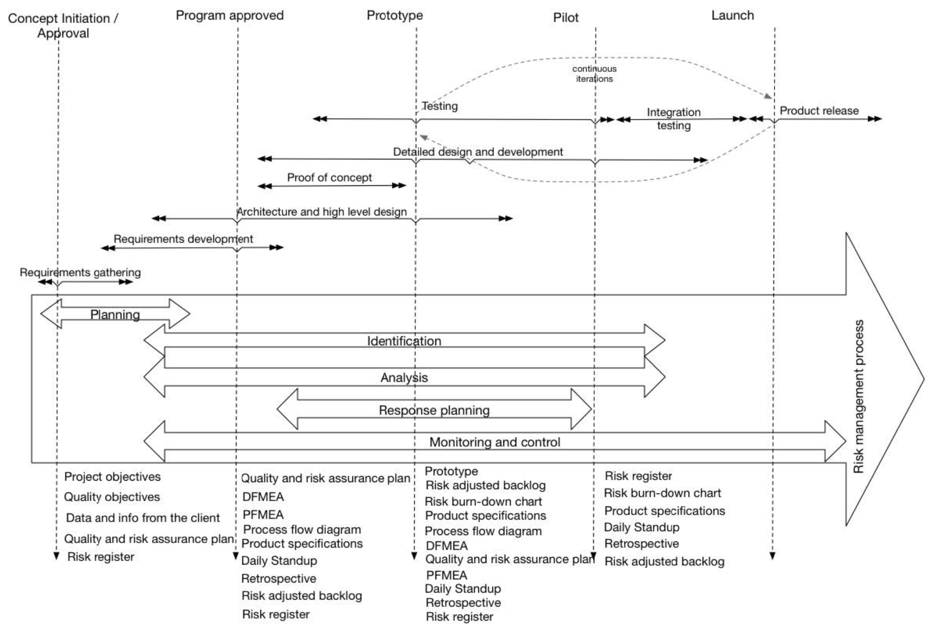


For mapping the tools and techniques applying risk management to the agile processes we utilized a standard risk management process as defined by PMBOK (Project Management Institute, 2013) and AMA Handbook (Dinsmore & Cabanis-Brewin, 2014), which we further adapted to the SDLC process, as an adaptation of risk management process to the agile environment in the context of the APQP framework (see figure 4):

1. Risk process and system planning
2. Risk identification
3. Risk analysis which includes:
    a. Qualitative analysis
    b. Quantitative analysis
4. Response planning
5. Monitoring and control

In order to evaluate the coverage of all the risk management tools deployed in the QHD HOQ, we plotted the tools and techniques to the adapted risk management framework. The results of the analysis are presented in Figure 4. The major constraint in streamlining the risk management process through an agile environment was to maintain it "lean" and "light". The primary concern of agile environments is to keep non-value adding activities as low in the priorities as possible, which includes documentation, supporting tools and techniques and heavy processes. The results in figure 5 also contain a number of agile specific risk management tools such as: Risk Adjusted Backlog, Risk Burn-down Graphs and the meetings "system" proprietary to Scrum methodology. The rationale behind including these into the proposed risk management process is that they are proprietary to the agile framework and complementary to the proposed risk tools and techniques.

Figure 4: Risk management process with associated tools and techniques



The stage gate framework allows for the tools and techniques to be implemented in a structured way throughout the risk management process. Furthermore, it enables the GO / NO GO decision making opportunities that agile environments generally lack. However, when coupled with an agile process (as presented in Figure 4), the deliverables and inputs from every stage are tightly coupled and repetitive. This is the same situation as for the adapted risk management process. In an industrial project setting, the risk management process steps are better established in time and more structured. The lengthened stages of the risk management process are primarily due to the iterative deployment characteristic of the agile environments. It is necessary in this case to iterate the risk management process (the identification – analysis – response planning – monitoring cycle) along with the sprint iteration. In order to plot the iterations we considered the more appropriate solution to graphically present it as a linear process, throughout the entire lifecycle – given though the effort at all times is not equal in intensity.

As for the tools and techniques assigned to all stages of the lifecycle, we can clearly also see that all the risk management process stages are well covered. In order to balance the risk undergone by iterating risk management tools and techniques, we introduced the "risk register" tool, a widely utilized risk management tool also proposed by the PMBOK. The role of this risk register would be to create, as per its name, a risk registry with the intent to pass on from stage to stage the risk management process know-how.

5. CONCLUSIONS AND FUTRE WORK

It is obvious that agile and Stage-Gate models are compatible and can bring great benefits to each other. Agile practices and processes can benefit greatly from the application of "lessons learned" in other, more advanced and structured domains. The adaptation of a rich framework such as APQP can be of great help to both managers and practitioners

An important area of future work is to properly validate the proposed risk tools and techniques roadmap, as the research proposition in Figure 1 depicts. As the scope of this paper did not allow even a partial validation, a case study application is required so as to provide a practical view on possible implementation possibilities, constraints and outcomes.

The scope of this paper was to add value to the risk management practices in agile environments by applying the lessons learned from the APQP model. However, throughout the research it became obvious that the stage-gate framework has the potential of scaling the agile processes organization wide, which is to create the setting for the organization-wide processes to interact with the project management agile processes.

ACKNOWLEDGEMENTS

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), ID134378 financed from the European Social Fund and by the Romanian Government.

REFERENCE LIST

1. Alharbi, E., & Qureshi, R. J. (2014). Implementation of Risk Management with SCRUM to Achieve CMMI Requirements. *I.J. Computer Network and Information Security , 11*, 20-25.
2. Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Highsmith, J., et al. (2001). *Principles behind the Agile Manifesto*. Retrieved 2015 from Agile Manifesto: http://www.agilemanifesto.org/principles.html
3. Boehm, B., & Turner, R. (2004). *Balancing Agility and Discipline.* New York: Addison Wesley.
4. Buyukozkan, G., & Baykasoglu, A. (2004). A survey on the methods and tools of concurrent new product development and agile manufacturing. *Journal of Intelligent Manufacturing , 15*, 731-751.
5. Chrysler Corporation, Ford Motor Company and General motors Company. (2008). Advanced Product Quality Planning (APQP) and Control Plan. *2nd edition*.
6. Conboy, K. (2009). Agility from first principles: reconstructing the concept of agility in information systems development. *Information Systems Research , 20*, 329–354.
7. Cooper, R. (n.d.). *Stage-Gate and Agile Development: Debunking the Myths*. From http://www.stage-gate.com/resources_stage-gate_agile.php
8. Cooper, R. (1994). Third-generation new product processes. *Journal of Product Innovation Management , 11* (1), 3-14.
9. Cooper, R. (2014). What's Next? After Stage-Gate. *Research Technology Management , 57* (1), 20-31.
10. Cooper, R. (2011). *Winning at New Products: Creating Value Through Innovation.* (4. edition, Ed.) New York: Basic Books-Perseus.
11. Cooper, R., Edgett, S. J., & Kleinschmidt, E. J. (2002). Optimizing the Stage-Gate process: What best-practice companies do—I. *Research-Technology Management , 45* (5), 21-27.
12. Dingsøyr, T., Nerur, S., Balijepallyd, V., & Brede Moea, N. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software , 85* (6), 1213–1221.
13. Dinsmore, P. C., & Cabanis-Brewin, J. (2014). *The AMA Handbook of Project Management* (4th Edition ed.). Amacom Books.
14. Dyba, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and Software Technology , 50*, 833–859.
15. Kettunen, P. (2009). Adopting key lessons from agile manufacturing to agile software product development—A comparative study. *Technovation* , 408-422.
16. Miller, G. G. (2001). The Characteristics of Agile Software Processes. *Proceedings of the 39th Int'l Conf. and Exhibition on Technology of Object-Oriented Languages and Systems* .
17. Noor, R., & Khan, M. F. (2014). Defect Management in Agile Software Development. *I.J. Modern Education and Computer Science , 3*, 55-60.
18. Project Management Institute. (2013). *A Guide to the Project Management Body of Knowledge: PMBOK Guide* (5th Edition ed.). Project Management Institute, Inc.