Quality Management techniques embedded in Agile Project Development

Raluca Dovleac[1,*], and *Andreea* Ionică[2]

[1]Management Department, Doctoral School, University of Petrosani, Romania
[2]Management Department, University of Petrosani, Romania

Abstract. It used to be hard to make your dream come true and start a business but now, thanks to the dotcom boom and the rapid development of emerging technologies everybody has the chance to become the next Steve Jobs. Quality management tools and creativity techniques gave birth to the startup trend that transforms a brilliant idea into an innovative product. A common practice amongst startups is Agile development. In this paper, after analyzing the Agile approach and the tools used to ensure quality for every stage of a product's development life cycle and the results of using the Agile approach looking out for advantages and short comings as well, we proposed a PDCA inspired tool which embeds both quality and creativity techniques to help developers ensure quality for each stage of a product's life cycle as well as help them better organize their activities to reach the desired goals. We validated this approach by implementing it to the case of the Vitraly startup.

1 Introduction

The history of the word quality encapsulates plenty of definitions and aspects as well as case studies. As of lately, quality in all its forms has become a major factor taken into account by all companies and small businesses as well. In this sense, and in order to help companies integrate quality into their everyday business lives, standards, procedures, tools and techniques come to their aid to ensure a standardization and ease of implementation.

An interesting aspect emerges however from observing the correlations between the implementation and maintenance of quality throughout a product's lifecycle and the project management process of that product.

Another important aspect to factor in is the way that the agile approach and quality merge together to ensure quality products and/or services are delivered.

In our paper, we outlined the important aspects of quality, project management and agile development as well as proposed embedding new quality management and creativity tools into the agile development thus ensuring better results.

[*] [*]Corresponding author: raluca.dovleac@yahoo.com

2 Literature Review

From its birth in the late 13th century, when craftsmen began organizing into unions called guilds, quality evolved and developed around the needs of manufacturing companies and production plants initially, only to become part of the modern companies objectives later on.

Given the fact that there are multiple understanding of quality and quality systems [1], different strategies for quality tools emerged as well [2]. When it comes to classifying the quality tools available today, there are several approaches to the topic, a fundamental one covering "The Seven Basic Tools of Quality" also known as 7 QC Tools. The 7 QC Tools developed in Japan in the 1950's by Kaoru Ishikawa when the country was undergoing some major revolution in the field of quality and it was inspired by seven well known weapons of Benkei [3]. The tools are often referred to as The Seven Basic Tools of Quality because they can be implemented by any person with basic training in statistics and can be applied to solve complex quality related problems [4]. They include: The Pareto Chart, Histogram, Process Flow Diagram, Control Charts, Scatter Diagram, Check Sheets and Cause and Effect Diagram [5]. Carpinetti [6] argued that the most important techniques are: Quality Function Deployment (QFD), Failure Mode and Effect Analysis (FMEA), Six Sigma, 5S, Design of Experiments (DoE) and Statistical Process Control (SPC). After 1977, The Union of Japanese Scientists and Engineers (JUSE) appreciated the following tools and techniques to be the most essential for quality improvement: The Affinity Diagram, The Relations Diagram, The Matrix Diagram, Tree Diagram, Arrow Diagram, Decisions Diagram and The Factorial Analysis of Data [7].There are those who debate on classifying quality tools according to the type of inputs they require into: qualitative tools such as: Flow charts, Cause and effect diagrams, Multi- voting, Affinity diagram, Process action teams, Brainstorming, Election grids, Task lists etc. and quantitative tools such as Shewart cycle, Control charts, Scatter diagrams, Pareto charts, Sampling, Run charts, Histograms. Another perspective of classifying the quality tools focuses on the reactive, proactive approach, dividing the quality tools in: Reactive tools - Graphics, Histograms, Checklists, Pareto diagram, Brainstorming, Election Grids, Solutions matrix, and Proactive tools - Relations, Affinity, Tree, Matrix, Alternative, Arrow diagrams, Principal components analysis [8].

When it comes to the evolution of the Agile approach, it can be said that it underwent some significant changes and stages in its development. Studies such as those of Abrahamsson [9], Highsmith [10] and Cohen [11] focus on the aspect of Agile developing as a reactive response to the outdated traditional project management methods. Unfortunately, this approach considers Agile to have appeared only in the last couple of years but the agile ideas have been around for longer than that. In their paper, Larman and Basili [12] noted the usage of iterative and incremental approaches since the 1970's.

In his paper, Beck [13] suggested that the Waterfall model was first developed to help asses and define user needs by noting the user requirements. The Waterfall model tried to fix the problem of changing requirements by establishing well defined requirements and not allowing any changes to be made to it, but as practitioners observed, the requirements just couldn't be well defined in one swoop. [14] Incremental and iterative techniques evolved as a response to the rigidity of the user requirements, and tried to improve upon the Waterfall model by keeping the main aspects of it but introducing the notion of overlapping increments to help reduce the development time. [15] Now, the requirements are also noted before the development stage begins, but they are broken into chunks of standalone functionality known as increments. The development of each such increment can be overlapped with the others, creating a "multitasking" effect and saving time. And while the incremental development aimed to reduce the allocated development time, there are models that provide better handling capabilities for changing requirements and risk management. [16]

In the table below, we have outlined the main aspects and differences between the traditional approach towards project management and quality and the agile approach to these aspects.

Table 1. Key aspect of traditional project management approaches and agile development

Traditional project management tools	**Agile project management tools**
Rigid. Team members must adapt	Flexible. Can adapt to team members
Abstract	Visual and intuitive
Works best with large teams	Works best with small and medium teams
Reporting is an element of major importance	Constant feedback is important
Works best with teams that share a work Space	Can adapt to remote teams
Interest in the final result	Interest in each sprint release
Well established quarterly/semester/annual Meetings	Daily meetings
Schedules a series of events	Schedules releases and product versioning
Works with critical path calculations	Works with burn down charts
Formal management structure	Informal management structure
Dependencies between tasks can be dealt with on the go	Dependencies between tasks must be dealt with as soon as possible as they directly affect product releases
Quality within	
Traditional Quality Management	**Agile development**
Focused on delivering project objectives	Focused on constant improvement of delivered products
Adapts to established requirements	Adapts to customer changing demands
Comprehensive documentations is mandatory	Working software is more important than documentation
Progress is monitored through reports and periodical meetings	Progress is monitored through daily meetings and results
Defines working products criteria	Works with user stories
Reactive response to change	Proactive response to change
Sustainable development	Sustainable development
Complex solutions	Simplicity is essential
Considers that best decisions are made by professionals	Allows teams to self-organize and gives freedom to team members in choosing architecture, requirements and design
Gives power to team leader	Gives power to team members

3 Project lifecycles and agile systems

The lifecycle of a project is the period of time and the necessary steps to be made for a project to be completed. When talking about the lifecycle of a project, the following aspects need to be considered: the type of work that has to be completed, the expected deliverables, the people involved and criteria for the approval of each phase [17]. The lifecycle of a project management includes five phases: 1.Project Initiation: The stage in which the project and its scope is clearly defined, feasibility studies take place to decide whether or not the project will be launched. The purpose and requirements of the project will be noted in a project charter or a project initiation document (PID). 2. Project Planning: In this stage, the objectives of the project are outlined, goals are established, resources are allocated and estimates about deliverables are generated. Some of the documents of this stage include: Work Breakdown Structures (WBS), Gantt charts, Scope statements, Milestones and Risk management plans. 3. Project Execution: During this phase the deliverables are developed, resources are assigned, the project manager directs and manages the project execution, a tracking system is set up, tasks are completed and verified and the project plans are updated if necessary amongst other things. 4. Project Performance/Monitoring: This phase focuses on ensuring that the performance and progress is on track with the established goals, and in order to do so, project managers will use tools such as the key performance indicators (KPIs). It is recommended to pick 2 to 5 such KPIs to accurately determine the performance. Another important aspect to factor in at this stage is the budget and checking if the project is within the planned budget at all times. 5. Project Closure: This phase takes place when the deliverables have been made and all the tasks have been completed. During this stage, review meetings will take place to identify what went well and bad during the project development and to get feedback from the team members as well. [18].

When it comes to Agile, the life cycle also known as System Development Life Cycle (SDLC) is composed of six phases instead of five as was the case with traditional project management: Concept Phase, Iteration 0/Inception, Construction, Transition/Release, Production, and Retirement [13]. 1. Concept Phase: In this phase the business opportunity is defined, the viability of the project is determined and feasibilities studies are conducted. 2. Inception/Warm up: The phase in which the funding is obtained and general support is shown, the team is working with the stakeholders in order to determine the goal of the project, the team is built, the architecture for the system is modelled, and project estimates are established. 3. Construction phase: During this phase, the team is maintaining a close collaboration with stakeholders and the rest of the team, priorities for tasks are set and quality is ensured, at the end of every sprint there are deliverables available for testing and the deliverables are analyzed and designed. 4. Transition phase: The transition phase is also known as the “end game” because in this stage the solution is released for production, the final tests take place, the documentation is written, and the system is deployed. Other activities that can be included here are: reworks – if the product has any defects and training – for both users as well as team members. 5. Production phase: In this phase, all the focus is on keeping the system running and helping users adapt to it and be able to use it (can include help desk activities). During this phase a common practice is assigning operational personnel for running and monitoring the system. 6. Retirement: The retirement phase aims to remove a system release from the production process, and that happens mostly because of the new systems emerging and replacing the old ones, but other reasons can include: the system is no longer needed on the market, it has become obsolete or it is no longer supported [19].

4 Integrating the quality management and creativity tools in the agile system development life cycle (SDLC)

Based on all the research conducted, and the points detailed above, we proposed a new approach for the life cycle of agile project management inspired by the famous Plan Do Check Act (PDCA) method. Similar to it, our model has only four stages / phases: Initiation; Production; Delivery; Refactoring.

In the Initiation phase the following activities are conducted: creating the business model, value proposition, crowdfunding, market analysis, planning sprints and releases, finding the right team members, and a minimum of documentation.

The Production phase focuses on activities such as: development and task completion through reaching the established sprints; continuous testing, inspection & checkups with stakeholders and clients.

In the Delivery phase, the final version of the product is released to mass market and stakeholders; feedback is gathered and decisions regarding the future of the product are made. As it can be seen from fig. 1, this phase is important because it affects the future of the product. Once the phase is completed, there are three options available: closure, maintenance or refactoring. For the purpose of this paper, and because it is closer to reality, we have considered refactoring as the alternative that will be chosen.

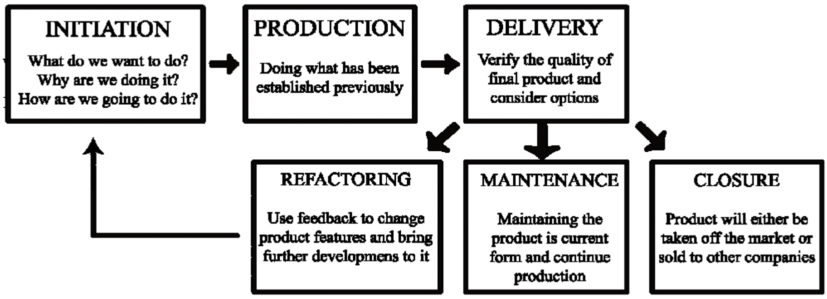
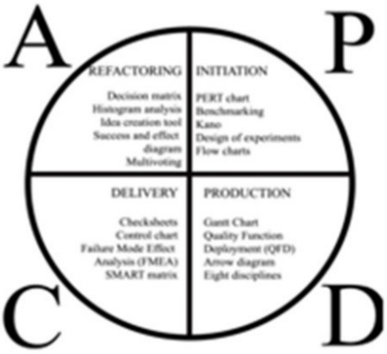

INITIATION
What do we want to do?
Why are we doing it?
How are we going to do it?
PRODUCTION
Doing what has been established previously
DELIVERY
Verify the quality of final product and consider options
REFACTORING
Use feedback to change product features and bring further developmens to it
MAINTENANCE
Maintaining the product is current form and continue production
CLOSURE
Product will either be taken off the market or sold to other companies

Fig. 1. Simplified System Development Life Cycle for Agile development projects

Optional Refactoring: The refactoring phase is perhaps the most particular one for the agile development because it links constant improvement with deliverables and it permits the team a flexibility in decision making. By going through the refactoring process, the team analyzes the feedback from the clients and decides what functionality to implement next taking into account their own ideas of course.

The main focus of this stage is deciding what to change about the current product and making sure that all the ideas are captured before making a decision. Once this stage is completed, the development process begins once more with the first stage of initiation.

This cycle can be repeated as many times as necessary in the agile development teams' environment.



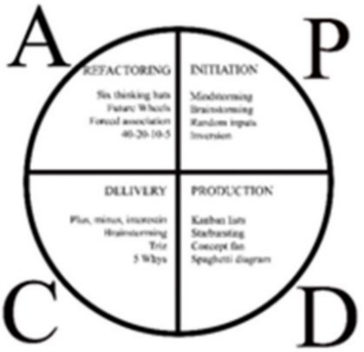


Fig.2. Traditional quality management tools agile SDLC

Fig.3. Creativity tools adapted adapted for the for each stage of the agile SDLC

Given the fact that quality is a must for both traditional project management teams as well as agile ones, in fig. 2 we described what quality tools can be used, and how they can be adapted for each stage of the development process to ensure quality for each step of the process. When it comes to quality assurance in the agile environment, it can be said that even though the traditional quality management tools suffice, new tools have emerged that can be used for the same purpose. Creativity tools have been used lately as problem solution generators; hence they can complement the traditional quality management ones into finding the best option for quality assurance and control for each stage of any project especially in the case of agile development projects. [15] In fig. 3 we noted the main creativity tools which can be used for this purpose and depicted how each of them can be applied for the stages of the development process.

5 Case study and results

The proposed model has been implemented for the case of three startups: Vitraly, Check4Green [20] and Timetrak to observe and register the results, improvements or drawback if they occured. Next we will describe the implementation process for each stage of the system development life cycle as well as the results we achieved so far. Vitraly is a startup which brings the smart element to a regular mirror, turning it into a highly customizable mirror that acts as a “double agent”. Besides displaying the world around it – like any other mirror would, it also displays widgets that the user wants such as: the clock, calendar, motivational quotes, weather and so on.

The team working on the product consisted of two engineers, two programmers and an economist. As was the case with the other two startups, for the project management of the Vitraly startup the free online project management platform called Freedcamp was used. For this, the team was created by inviting the members of the startup to join the Vitraly project group. In the initiation phase, the team held a brainstorming session combining this technique with the “Random Inputs” creativity techniques which requires the users to find as many correlations between unrelated items. Even though, a rough idea of a smart mirror was formulated already by the team during casual talks and interactions, a certain list of features and product capabilities wasn’t clearly defined. During the brainstorming session, plenty of features for the smart mirror were identified such as for eg: it should display the temperature outside; it should display personalized messages that the user wants; it should allow the user to set alarms; it should turn off the widgets when the user is not around and display nothing, for power saving purposes, etc. All these features were noted down by the product owner and were later translated into user stories and tasks for each team member. During this stage, the team noted the lack of support for ideas, in the sense that the ideas captured during the brainstorming session couldn’t be stored on the spot in the Freedcamp platform unless the team members inputted them in the discussion board afterwards which was time consuming and useless. After the ideas were selected and the team clearly defined the user requirements, these were noted on the platform by the product owner and later, tasks were derived from them by the scrum master and assigned to team members whilst defining the deadlines and allocated resources as well. Still in this stage, in order to better and easier define the business model, the team used the 40-20-10-5 method of problem solving which states that when dealing with a problem, a person must explain it in up to 40 words, then cut it down to 20 words, then 10 and lastly, to 5 words. These are the root cause of a problem and also the solution to that problem [21].

During the production phase, the product owner made sure that everything was going according to plan and with the help of the Kanban list view of the tasks that the Freedcamp platform offers, also supervised in real time the completion of the tasks and if these were in schedule with the established timeframes. Still during this stage, constant small releases were made to the public, more specifically, whenever a new widget was developed it was already deployed in order to gather the end user’s opinions and adjust accordingly. Freedcamp didn’t offer any kind of support for this activity, so once again, the changes in both customer demands and product features couldn't be noted only written as new tasks whilst marking the old tasks as completed or simply deleting them. Either way, a proper time estimate of how much time a task took if it was affected by the end user changes couldn't be made using this platform.

As soon as the product was finished it was launched on the market - encapsulating the delivery phase, through the help of a crowdfunding campaign in order to both get funding for further developing the product and producing more of it as well as to maintain a high level of feedback received from the customers who are more actively involved in the case of crowdfunding campaigns. Given the fact that people loved the idea but wanted improvements such as: making it available in smaller dimensions, more widgets and layouts that can be customized from the mobile phone, the team decided to step into the refactoring phase and decide what to deliver next though the help of the creativity methods noted in Figure3. Throughout the development lifecycle of this project as well as the other two: Check4Green and Timetrak, the usage of the free project management platform called Freedcamp proved to have some gaps in managing projects which are using the Agile approach. In order to fill these gaps and reach the established goals, the development team had to use a combination of quality management and creativity tools described above and noted in fig. 3.

5 Conclusions

Keeping in mind the principles of the Agile Manifesto, especially the one referring to less documentation and more valuable working code, we have designed a system development life cycle that is easier to understand, apply and monitor, inspired by the Plan-Do-Check-Act (PDCA) cycle. Given the fact that a product developed by an agile team does not necessary end after the delivery to the customer and that it can go through changes both during the production phase and the post-delivery one, the whole process can become cyclic a couple of times rather than having a well determined beginning and ending. The similarities between this and the PDCA cycle allowed us to define a system that depicts the lifecycle of an agile development environment more closely to reality and in accordance to the Agile Manifesto itself.

In order to use this model for a project, one must first define each stage of the development process and and know which stage they are currently facing then apply the following for each step. In the Initiation stage, the following quality tools can be applied: PERT chart; Benchmarking; Kano; Design of Experiments and Flow Charts. As for the creativity tools, we can have: Mindstorming; Brainstorming; Random Inputs and Inversion. In the Production stage, the quality tools used in the model are: Gantt Chart; QFD; Arrow Diagram; Eight disciplines and the creativity tools are: Kanban lists; Starbursting; Concept fan and Spaghetti Diagram. The Delivery phase encapsulates quality tools such as: Checksheets; Control Charts; FMEA and SMART matrix and the creativity tools as: Plus, minus, interesting; Brainstorming; Triz and 5 Why's. The last optional stage named Refactoring, quality tools such as: Decision Matrix; Histogram analysis; Idea Creation Tool; Success and Effect Diagram; Multivoting are used along with creativy tools such as: Six Thinking Hats; Future Wheels, Forced Associations and the 40-20-10-5 method.

Besides all this, keeping in mind the element of flexibility and creativity that the agile trend brings, we have incorporated creativity techniques for each stage of the development process so that it can be in line with the whole principle of this trend and adapt easily to any sort of agile development team. Throughout the whole production process and at the end of it, the promised quality will be monitored and delivered with the help of a combination of the traditional quality management tools integrated with the modern creativity tools used for problem solving purposed and quality assurance.

References

1. P.B.Crosby, *Quality is Still Free: Making Quality Certain in Uncertain Times* (McGraw-Hill,
2. 1996)
3. K. Liang, Mod.Appl.Scient, **4**, 9 (2010)
4. K. Ishikawa, Introduction to Quality Control (Springer Netherland, 1989)
5. A.Q. Press, *Seven Basic Quality Tools* (ASQ Quality Press, 2010)
6. R.H. Fouad, A. Mukattash, JJMIE, **4**, 6 (2010)
7. L. C.R. Carpinetti, T.Buosi, M.C. Gerólamo, Business Process Management Journal, **9(**4), 543-554 (2003)
8. A.C. Ionică, *Managementul calităţii* (Universitas, 2006)
9. R. Moţoiu, *Ghid pentru managementul calităţii şi calitatea managementului anilor 2000* (Fiatest, 2010)
10. P.Abrahamsson, O. Solo, J.Ronkainen, J.Warsta, VTT technical Research Centre of Finland (2002)
11. J.Highsmith., *Agile Software Development Ecosystems*, (Addison-Wesley Longman Publishing Co., 2002).
12. D.Cohen, M.Lindvall,. P.Costa, Advances in Computers, 1-66 (2004)
13. C.Larman, V.R.Basili, IEEE Computer Society **36** (6), 47-56 (2003)
14. K..Beck, *Extreme Programming Explained: Embracing Change*, (Addison–Wesley, 1999)
15. K.Beck, *Embrace change with extreme programming*, 70–77, (IEEE Computer, 1999)
16. A.Cockburn, Cutter IT Journal, 6–12 (2002)
17. Cockburn A., Highsmith J., IEEE Computer, 120–122 (2001)
18. T.Kendrick, *Identifying and Managing Project Risk: Essential Tools for Failure-Proofing Your Project*, (Amacom, 2003)
19. R.K.Wysocki, *Effective Project Management* (Wiley, 2014)
20. M.Cohn, *Agile Estimating and Planning* (Prentice Hall PTR, 2005)
21. R.Dovleac, A.Ionica, M.Leba, Proceedings of Review of Management and Economic Engineering, 32-39 (2016)
22. A.Ionica, M.Leba, E.Edelhauser, Transformations in Business & Economics, **13**, 2B, 697-716 (2014)