

Project Management Tools for Agile Teams

Alexandra MIHALACHE

Faculty of Cybernetics, Statistics and Economic Informatics

Bucharest University of Economic Studies, Romania

alexa.mihalache@yahoo.com

The 21st century started as the era of agile methods and technologies which combine adaptable and highly flexible methods designed to help teams to develop applications faster and safer. The challenge is to adapt just enough technology to enhance the performance of teams which are using agile methods and to avoid going back to weak workflow ideas of the industrial era. In this paper, I will present some relevant agile tools that could improve every software development project and I will also mention the features and criteria used for evaluating currently existing tools. Finally, I will propose a classification model to the appropriate agile tool selection, but keep in mind that in order to be successful, organizations and teams do need to deal first with the issues that accompany product complexity and growing teams.

Keywords: *project assessment, agile tools classification, unpredictable situations control*

1 Introduction

Agile methods are undeniably one of the most important and recent developments in software engineering. Over the past 30 years they have highly boost success rates in software development, motivation and productivity of IT teams and also improve agility to market and quality. Spontaneous market changes have changed the business plans for most companies, so they had to shorten the delivery of a product and to be more responsive to the expectations and demands of customers. Agile development methodologies are very useful tools, representing a real support to the development of software processes, being able to adapt according to the new market trends. Implementing agile methodologies have a huge impact on the discipline of business analysis in general, since they put in a forefront a new method of approach and prioritization of requirements within a team which is implementing an IT project.

Successful management of an agile project depends heavily on the process of defining requirements and developing smaller steps which means a closer collaboration with the whole team along SDLC (Software Development Life Cycle) of the project.

Migrating from a perspective of a traditional project approach to an "agile" will lead to a redefinition of functional roles within a

project team in a different manner.

Agile is about innovation and nowadays most companies operate in highly dynamic environments, which mean they need tools that can adapt quickly to changes. They do not need just new products and services but also innovation in their business processes. Companies which create an environment in which agile dominate, find that teams can make innovations ideas really happen in a faster way, so in this paper I will compare the most used agile tools which lead businesses to success, providing comparisons between them and I will also analyze the changes of business processes in a Romanian bank, due to the adoption of the PM agile tools.

2 Literature Review

The aim of this section is to provide the background information of the paper topic. The main purpose is to establish a solid base of the existing literature, and to investigate questions like: "Why do you need an agile tool or how can it change your business?" or "How can agile development tools improve IT project success?"

Relevant data regarding my research about the use of agile methods is represented by one of the most important tools in social science: surveys. They are often used to collect data about management trends, such as agile methods, or about attitudinal data,

representing a form of quantitative research. Surveys have been used to collect key information, such as the size of the project, the number of people trained to use agile methods, years of experience, the types and practices being used in an organization, its size and revenues, which are also very important. The impact of agile methods on productivity, costs, quality, customer satisfaction or cycle time must be frequently measured and improved. Compared to traditional methods which usually lead to cost escalation, poor requirements and quality, high maintenance cost or project cancellation, agile methods lead to satisfaction of customer needs and successful project completion. Therefore, one of the most frequently asked questions is “Do agile methods lead to project success?” is going to be answered through the most relevant works I read for this research. This section focuses on agile methods in general which are being presented as follows. In 2006, Microsoft took a survey of agile methods among 10% of its software developers, even though its own development practices are considered to be the progenitors of agile methods. About 32% of the sampled (2 800 developers) said they used agile methods, this fact being a surprise to the researchers who planned the survey, because they were unaware of the link between an existing research by MIT where it was mentioned that about 21% of Microsoft’s developers were estimated to be using Scrum. MIT (Massachusetts Institute of Technology) conducted the research mentioned before, concerning software development practices from 1993 to 1995 and published the results in numerous journal articles and also as a textbook. Even though it was confirmed that Extreme Programming’s practices such as continuous integration, test-driven development and release planning, came from Microsoft, one of the most frequently noted factors was that agile methods worked well for individuals, teams, and groups, whereas the least frequently reported factor was the attitude. There were also two benefits reported mostly: improved communication and quick releases, which go hand in hand with a certain

business success.

Scott Ambler of both AmbySoft and IBM has written a lot of books on the subject of software development and agile methods related topics. One of his most important goals was to measure the rate of adoption: how many people are using agile methods and what are the effects on their business projects?

Ambler’s surveys were focused on topics such as type of method/ practice used, organization size and knowledge percent of agile methods. He was interested about the impact of agile methods on quality, costs, productivity, customer satisfaction and project success, but he also focused on iteration length, team size, number of projects that used agile methods and performance.

In his first survey, in the early 2008, 65% of 4000 respondents came from small organizations, about 10% considered themselves experts and only 41% had adopted agile methods. The most often used methods were Extreme Programming and Feature Driven Development and only 10% of the respondents reported significant benefits.

In the next year’s latest survey he decreased the number of respondents to 700. Almost 70% of them reported they have adopted agile methods and that 73% of projects were successful thanks to agile methods. Improvements in quality, costs, productivity, and customer satisfaction as a result of agile methods doubled from 10% to 20%. Ambler’s later work focused on the scalability of agile methods because they were considered to work only for small and simple problems, but he realized that agile methods were actually ideally suited for large and complex problems. In a rather startling role reversal, that was the moment researchers were also started to realize that traditional methods may be ideally suited for small and simple problems. [2]

In 2009, according to „The Business Value of Agile Software Methods” agile tools were considered to be a better alternative to traditional methods which do not focus on the real value of the business, but on the other hand the agile ones were criticized for not supporting the complete life cycle of a system. By placing traditional and agile methods in

their proper context, the authors proved that the agile methods are just enough, in time and right sized approaches for maximizing the business value by managing the development of innovative software products. [2]

In 2012 – The results of the 6th Annual State of Agile Development survey, sponsored by VersionOne, show several interesting trends among the agile community. Based on a sample of more than 6,000 persons, the survey findings show that software tools for agile management are in high-demand among respondents with multiple agile projects, and their use contribute to many successful agile projects because they help users to execute complex projects in an intuitive and time-effective manner. Their management creates suitable deadlines according to the project details that users submit in the software and they do not have to depend on inaccurate guesses. According to survey results, there is a huge correlation between the number of agile projects, the use and benefits from a project management tool, built especially for agile. Organizations with 25 % or fewer agile projects tend to rely on traditional project management tools to manage their projects, but the most of them are more likely to use a purpose-built agile management tool like VersionOne. [3]

There are many agile tools designed especially for management, collaboration, development or support, including agile technologies which are flexible and adaptable to constant changes. They focus on interaction and communications which are considered to be much more important than tools and technologies. Agile management tools are very important to project teams, allowing them to execute projects using Scrum and Extreme Programming in an efficient way, following predetermined rules. There are a lot of commercial and open source workflow tools, ranging from sophisticated web-based applications to simple and specialized utilities. The most popular tools are VersionOne, Rally and Jira which provide support for both release and iteration planning. One of the most basic tools used by agile teams is the whiteboard, which is being

known as an information radiator. Agile collaboration tools build solid relationships between customers and teams. They also help teams to communicate efficiently, collaborate and share needed information to implement the user stories. Like management tools, there are also numerous commercial and open source collaboration tools, some important ones supporting video conference feature to make distance unimportant. The telephone is one of the most basic tools used for this purpose and social networking sites play also an important role regarding communication, giving the businesses opportunities to share personal data and promote their projects and brand. The development tools are designed for coding and testing, helping teams with the technical aspects of the software. Commercial tools provide compilers and interpreters for different programming languages, databases or graphical user interfaces. One of them is Visual Studio, developed by Microsoft, which is a proprietary system running only on Windows. On the other hand, Eclipse, an open source tool, support multiple platforms, provide simple modeling tools, graphical editors and reusable frameworks. The focus is to create working software and for this are needed robust environments, whether commercial or open source. Going to agile support tools, they help teams to build and deliver working software every 14 to 30 days, depending on how long iterations are. Many of them provide direct support for the creation of the working software. Automated testing frameworks and open source utilities are the most commonly used tools, commercial solutions being also very important because they provide well-integrated code analysis, testing and build automation. [2]

Returning to Ambler, in a recently survey which ran from January 31 this year to February 19 under the title “February 2017 Agile Mini-Survey”, 78% of the 180 respondents reported that their agile teams are being governed in some manner, but they are not totally governed in a lightweight or agile manner. To produce positive results, agile manner must be adopted because traditional approaches to governing agile teams backfire,

producing negative results. Regarding deliverables, for traditional governance strategies they can be easily faked by project teams without the risk of being caught, while deliverables for agile governance strategies are difficult to fake in practice. [5]

Surveys, especially the annual ones by AmbySoft and VersionOne, were very important to my research, representing important studies for measuring agile methods. They proved that customers' dynamic demands and the increasing intensity of global competition determined the organizations to adopt agile principle and use special tools, but none of the mentioned research provided a methodology for right tool selection.

3 Methodology

Agile is mainly a set of values and principles, and unless one can imagine ways of organizing a project team around these values and principles we can say that person is not agile. This is very important to understand whereas many are rigid in supporting a unique way to be agile. In one sense, agile methods are an outgrowth of the plentitude of rapid development methodologies emerging from the 1980s. From my point of view, each has its own vision of what agile means and often it is hard to share this strategy with others, showing a high resistance to change when the direction of agility is not anticipated.

Since the research is about agile project management tools, in this section I will focus the discussion on tools that would provide a full capability for handling large and complex enterprise projects, outlining their major features, after I read and analyzed many surveys, papers and after I talked to some of my colleagues who adopted agile few years ago in their department.

I. Tool Features

Agile project management tools are fundamentally different from traditional ones, because we talk about dynamic and adaptive approach, based on the concept of a continuous management flow of the project. As agile projects become larger and more complex, it becomes essential to use

specialized tools to manage and share information quickly and efficiently. There are a lot of commercial and open source management tools, ranging from sophisticated web-based applications to simple and specialized utilities. To keep a better comparison the following **features** maximizing business value were analyzed:

- **Agile reporting and metrics**
 - Time tracking and projection
 - Quality assurance
 - Easy-to-understand progress reports for stakeholders
 - Integration with the company's existing system
- **Communication**
 - Communicate updates within teams
 - Share task lists, assignments and feedback
- **Project assessment**
 - Identify and remedy project incidents
 - Evaluate constantly performance
 - Estimate financials

II. Comparison Criteria

Based on my co-workers feedback and the plenty of surveys and articles I read to prepare a list of the most important criteria that agile management tools must have to satisfy agile teams, and I present them as follows.

II.1. The business needs

There are many good agile project management tools in the market, so this is making hard the selection of a suitable one depending on every team needs. So the first step is to define the business needs.

II.2. The costs

Some tools are free, some are paid, and some are called by the new business model "Freemium", giving the users a trial or a distilled version of the software. There are many vendors which offer great free trial plans that teams can use to check out the software's features and benefits. I think this is a very good idea before buying the product, because the free trial period gives ample time to find out how the tool can boost a company's growth.

II.3. Progress visualization

This feature makes it easy to see in real time the status of projects and it also makes it simple to identify difficulties. If this feature is

versatile, it means that the team is not limited by the software’s visualization structure, such as a Gantt chart.

II.4. Issue and time tracking

Issue tracking works perfectly together with project visualization and allows the team to easily identify the problems of a project, as well as the completed parts of it. Time tracking is also important, helping the team to deliver working software in time. This features are used commonly by software development teams and help them to manage revision-related work.

II.5. Collaboration

Depending on the nature of the project and team it is very important to select suitable collaborative tools. This feature eliminates email chains and makes it easy to find the information needed faster. A good agile software has robust collaboration tools for centralizing documents attached to projects on the board. In particular, employee collaboration is vital in agile.

II.6. Estimation

Deadlines must be respected! Estimation is very used by project managers to set accurate requirements, to calculate how long an iteration takes and to update estimates after every release. Using a tool with this feature implemented, project managers can gather accurate estimations to generate a base of historical data which they can use to forecast future projects.

II.7. Project Portfolio

Whereas agile can be difficult to scale because of its focus on individual contribution, team collaboration and constant changes, this issues can be solved if project managers can quickly access the entire portfolio of projects.

II.8. Customer Portal

Designed for customers, it is used for detailed feedback based on which tasks can be planned. A customer development portal helps project managers carry out software walk-throughs and development teams also benefit as they can ask for feedback soon after a release. [5]

4 Results

In this section I will present a detailed comparison between some relevant agile tools that could aide in every software development project. In the first table are presented four commercial agile tools, while in the second one are presented spreadsheets and simple tools, all of them trying to give you answers to questions like: “Do development teams need to use expensive tools to produce good software?”

or “What is the key for selecting the appropriate tool for your business?” considering some of the following features:

- Iteration planning and project tracking
- Basic/ advanced reporting & dashboards
- Release and sprint planning
- Acceptance and regression test management
- Epic management and agile portfolio management
- Centralized story & defect management
- Team collaboration and customer idea management
- Integrations and multi-project support
- Sprint review and retrospectives
- Customizable workflow and views
- Burndown, velocity and test trends

Table 1. Agile tools comparison- Commercial modern tools

	Commercial modern tools			
	JIRA	VersionOne	Rally	Visual Studio Team Services
License	Proprietary/ Free community licenses for open source and academic projects	Proprietary/ hosted	Proprietary/ Free trial	Proprietary/ Commercial
Price	Multiple price points/ Free trial	Multiple price points/ Free trial	Multiple price points/ Free trial	Multiple price points/ Free trial/ Free
Platform	Web-Based/ Installed	Web-Based	Web-Based	Web-Based/ Installed

Intended Users	Freelancers, Large Enterprises, Small-Medium Businesses, Non Profits, Public Administrations	Freelancers, Large Enterprises, Small-Medium Businesses, Non Profits, Public Administrations	Freelancers, Large Enterprises, Small-Medium Businesses	Large Enterprises, Small-Medium Businesses
Backlog Management	Full support	Full support	Full support	Full support
Task board view	Yes	Yes	Yes	Yes
Iteration burn down chart	Yes	Yes	Yes	Yes
Epics (hierarchy of backlog items)	Partial support	Full support	Partial support	Partial support
Release and Iteration Planning and Tracking	Partial support	Full support	Full support	Full support
Multiple products/ projects	Full support	Full support	Full support	Full support
Portfolio planning	None	Full support	Full support	Full support
Test Management (Acceptance and Regression)	Partial support	Full support	Full support	Full support
Automated Notifications of Changes to System Assets	Email	Email	Email	Email
Tracking resources, time, tasks	Yes	Yes	Yes	Yes
Integration, API(s), SDK	Yes (REST API)	SDK:.Java, .NET, Python, Javascript	SDK:.Java, .NET, Ruby, Nodejs	SDK:.Java, .NET
Support	Email/Phone Community Website	Email/Phone Community Website	Email/Phone Community Website	Email/Phone Community Website
Service	None	Training and Certification	None	Training and Certification

Table 2. Agile tools comparison- Spreadsheets & Simple tools

	Spreadsheets		Simple tools
	Google Suite	Microsoft Project	Whiteboard/ post-it
License	ICU license	Proprietary/ Commercial	None
Price	Multiple price points/ Free	Multiple price points/ Free trial	Almost free
Platform	Web-Based	Web-Based/ Installed	None
Intended Users	Freelancers, Small Business	Large Enterprises, Small-Medium Businesses	Small-Medium Businesses
Backlog Management	No	Full support	Manual
Task board view	Yes	Yes	Yes
Iteration burn down chart	None	Yes	Manual
Epics (hierarchy of backlog items)	None	None	Re-stick some notes
Release and Iteration Planning and Tracking	Partial support	Partial support	Manual
Multiple products/ projects	Partial support	Full support	Partial support
Portfolio planning	None	Full Support	Manual
Test Management (Acceptance and Regression)	Partial support	Partial support	Poor support

Automated Notifications of Changes to System Assets	No	Email	No
Tracking resources, time, tasks	Yes	Yes	Manual, asking each person
Integration, API(s), SDK	SDK:Java, .NET	SDK:Java, .NET	No
Support	Forums	Email/Phone Community Website	Direct communication
Service	None	Training and Certification	None

There are also some key questions you will need to ask before to make the right choice. Considering the following questions: “How large is your team?” “Is the tool scalable?” “How much does it cost?” “Are there remote resources?” “Is the solution hosted or will you maintain the infrastructure behind it?”

The answers to these simple questions will not only drive your organization to the ideal tool but will also help determine the cost.

5 Summary of Findings

In order to be successful, organizations and teams must deal with different issues that accompany growing teams and product complexity. An important key is represented by the agile tools, because they help track productivity, (both individual and team), have the ability to respond quickly to change, use remote/virtual resources and create a strong collaboration with the customer. The use of agile tools gives a clear line of sight to the customers’ priorities and what they can expect at the end of the sprint/iteration.

Just because an organization uses an agile tool does not guarantee that it will be successful because it adopted agile methodology. Agile is based on simplicity and the tools should reflect this especially, because there are many applications which offer dozens of features that teams may never use. As an organization, the migration to an agile environment can seem daunting. Combining this with the selection of an appropriate tool it only brings more challenges but, at the same time it highlights the importance of the right tool, which may be the defining factor for a successful transition to agile.

In the first section of my research I mentioned that I will define the changes of business processes in a Romanian bank, due to the adoption of the PM agile tools. For this

purpose I talked with a Product Owner within the company I work. After he explained me that in the past they used to develop projects in a traditional way, using Waterfall methodology, a linear-sequential life cycle model, I asked him “Why agile and what was important in the process of choosing the appropriate tool for business?”

Regarding the first half of my long question, he explained me that in agile methodologies, leadership encourages teamwork, face-to-face communication and accountability in a stronger way. Business stakeholders and developers must work together to align the product with customer needs and company goals. We talk about constant change which is embraced perfectly by agile methods, and high-quality delivery, because the goal of each iteration is to produce a working product. They had to choose the right methodology to fit their projects, it was not a transition just to be in trend.

After many meetings and analysis between the features of about 15 tools they decided that JIRA fits their needs the best. I will mention the pros and cons they took in account four years ago, when they adopted agile:

PROS

+ Bug tracking: you can track the flow through the environments the bugs have passed (development/testing/development, etc.)

+ Collect statistics and make predictions according on how things worked. It can provide an estimate of the sprint/project completion (if people estimate/log the time correctly and honestly)

+ Cost optimization tracking. The project evolution it is very important because it lasts until all user stories are developed and bugs are solved. JIRA gives you the right information about what was wrong and what

you need to optimize.

+ Versioning: keeping versions of objects so every change can be seen by everyone who is part of the project (developer, tester, business analyst, scrum master, product owner)

CONS

- The management of the production process is not very well illustrated. He referred that

you do not have a very good image (as a kind of graphic) on what is happening in the project. For example, now all the focus is on the development of the requirements (it's a peak in the graph), then the graph line drops and moves to testing, followed by a chart increase in the test area (because all the effort moves to the test)

Table 3. Commercial agile tools: pros and cons

	PROS	CONS
Jira	Big community, Multi –language support, 600+ plugins and add-on, mobile	Poor backlog, sprint management tools Lacks burndown and resource reporting
VersionOne	Free trial for up to 10 users Supports collaboration for cross-functional teams Robust planning abilities; tracking Epics, Stories and Projects	Complex User Interface Not mobile ready Higher learning curve needed to understand all features
Rally	Supports collaboration for cross-functional teams Provides story and feature roll-up for enhanced program and portfolio management Includes integrated defect management	Complex User Interface Requires additional process for linking stories and features to higher-level portfolio items Lacks configurable out-of-the-box reporting
Visual Studio Team Services	Useful features for managing agile processes	It is desirable to use other Microsoft tools for development
Google Suite	Good tool for small teams and small processes. No installation of hardware or software is required	Manual work
Microsoft Project	It's foolproof - as long as you enter everything correctly and it requires no upfront infrastructure costs, offering flexible subscription plans, helping you keep your technology costs low	The software is pretty expensive which makes it difficult for smaller companies to afford and if you have a previous version of Office and you have purchased a newer version of Project, these are not compatible. Manual work

- The difference between a new user story and an old one from another sprint. You cannot differentiate user stories appurtenance. If you start a sprint and you do not finish to develop all the user stories, they will be part of the next sprint, but in JIRA you cannot make a difference which are from the previous sprint/s and which are new.

Even tough JIRA is a powerful tool, the product owner told me they never gave up on simple tools like whiteboard, high performance tree and ceremonies which are very useful.

o Whiteboard

There is no longer human-computer interaction, because you see physically how the tickets/post-it are moved (including user-stories/bugs) on the board from one category

to another according to the used methodology: Scrum (to do, in progress, done) or Kanban (backlog, selected for development, postponed, reopened, ready for test, done).

You became more motivated and stimulated (“Look X has moved 3 tickets today! I'm going to do the same.”). Your relationship with your team workers is improved because discussions/debates are started on different issues.

o High performance tree

It is another visual management tool, with whom each person in the team express his/her mood, sticking a leaf to the tree according to it. Thus relationship comes in: "Are you okay today? What needs to be done? "

o Ceremonies

Sprint meetings (e.g. daily meeting, grooming, planning, sprint review, retrospective)

6 Conclusion

After all I have read and all the information I have gained I am sure that there is a tool for any agile project being worked on, but the key is to figure out which one is right for your organization. Choosing an agile tool suitable for managing your particular processes is no easy task. All of the products mentioned in the table from above (apart from Google Suite) provide a free trial period time, where a team can try out the tool functionalities, check how it works in a real project and how it fits the project's needs, and also how it improves the development and planning processes.

Regarding the tools comparison, VersionOne and Rally provide the highest level of features for managing multiple agile projects for established and mature teams. These functionalities help clients to keep track of the team progress and monitor what obstacles appeared, giving the possibility to overcome them by providing important reporting mechanisms. Nowadays, developers in many teams work remotely and these tools also make working in such teams as comfortable as possible. Atlassian Jira is appropriate for project management, bug tracking and it has the availability of a large amount of plugins, which is a high benefit, being one of the most used and appreciated tools. It is very important to say that its add-ons let users customize and optimize the management process to fit their exact needs.

Visual Studio Team Services is also a

powerful integrated tool, having a lot of effective features easy to use. In the circumstances that Microsoft is the ancestor of agile methodologies, the company improve a lot in their development process and functionalities. Google Docs and Microsoft Project are both effective tools for small teams with relatively simple processes. Their main advantage is ease of use and handling.

In the table below I highlighted the pros and cons of the tools described, letting you taking the good decision depending on how they fit or not your business needs.

References

- [1] Rigby, D. K., Sutherland, J., & Takeuchi, H. (n.d.). Embracing Agile. Retrieved March 22, 2017, from <https://hbr.org/2016/05/embracing-agile>
- [2] Rico, D. F., Sayani, H. H., & Sone, S. (2010). Summary of agile methods. In *The business value of agile software methods: Maximizing ROI with just-in-time processes and documentation* (pp. 96-97). Boca Raton, FL: J. Ross.
- [3] State of Agile Report: Agile Trends - 2017 Agile Survey. (n.d.). Retrieved April 02, 2017, from <http://stateofagile.versionone.com>
- [4] 2017 Agile Governance Survey Results. (n.d.). Retrieved April 04, 2017, from <http://www.ambyssoft.com/>
- [5] Cobb, C. G. (2015). Agile Project Management. In *Project Manager's Guide to Mastering Agile: Principles and Practices for an Adaptive Approach* (pp. 139-149). John Wiley & Sons.



Alexandra MIHALACHE graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2015. She pursued her studies in Economic Informatics and this year she obtained her master's degree at Bucharest University of Economic Studies. Her interest focuses on IoT and its applications in different fields.