Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 64 (2015) 529 – 536

Procedia Computer Science

Conference on ENTERprise Information Systems / International Conference on Project MANagement / Conference on Health and Social Care Information Systems and Technologies, CENTERIS / ProjMAN / HCist 2015 October 7-9, 2015

Blisstrail: an agile project business case study

Bruno Antunes, Diogo Santos, Eurico Lopes*, Filipe Fidalgo, Paulo Alves

Escola Superior de Tecnologia, Instituto Politécnico de Castelo Branco, Av. Empresário, Campus da Talagueira 6000-767 Castelo Branco, Portugal

Abstract

Today's software development environment dictates speed, flexibility and a people-centered focus. Virtual teams represent a new organizational form that does not provide for same proximity collaboration. In this paper we propose a solution approach of a virtual team, as well as show some examples of using the Microsoft Visual Team Foundation Server 2013 tools to address these challenges.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge

Keywords: Scrum; MS Visual Studio Team Foundation; Travel planning.

1. Main text

Blisstrail “Never lose your way” is the project on which this paper will focus. This project will focus mainly on travel planning, the integration of online stores adherent establishments.

The project will be divided into two parts, the first focuses on an innovative portal, the second focuses on an application for smartphone that complements the portal created in the first point, adding new features to this project.

The portal will allow users to make a rigorous and thorough planning a trip (with or without touristic purposes), at the same time releasing products that are available in SMEs (Small Medium Enterprises) in the area covered of travel, this last point is guaranteed due to the creation of a model of online store, for shops adherent to the project. This point

* Corresponding author. Tel.: +351-272-339340; fax: +351-272-339-399.
E-mail address: eurico@ipcb.pt

1877-0509 © 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of SciKA - Association for Promotion and Dissemination of Scientific Knowledge
doi:10.1016/j.procs.2015.08.558

is very important since to come potentiate local economies, benefiting shopkeepers, customers, and the general economy of the region, providing information about these places and the business world.

The smartphone application aims to help, guide, complementary to optimize according to all that surrounds us, through a feature "geo-location" which will allow information to places of interest, shops and their products, etc... Through a simple action like pointing the camera of your smartphone for a given location.

In a survey by it was found that most respondents always or almost always work in virtual teams, despite that, not all find it advantageous, there are respondents who feel confused and dominated with the mixture of collaboration technology.

You can come to pass a new generation of technology that can make dispersed teams do more productive work. The Forrester Research Henry Dewing says that clarified users are using virtual services rather than meet in physical sites.

Given the fact that the development team as well as key partners, may not hold meetings and / or development of the face-to-face project, we chose to follow a set of management behaviors of virtual teams, such as:

- Setting ground rules for managing virtual communication
- Alignment of all personal and professional objectives
- Reinforcement relationships between all elements of the form team achieve better results in teamwork

Regarding the methodology used in developing the project methodology was "Scrum". This is an agile development methodology that will recognize the existence of problems and unanticipated changes throughout the process of software development. The main methodological feature is the focus on customer value, always seeking first perform the tasks that matter most.

The Scrum methodology is structured in several cycles, in each of these cycles assume to achieve a set of tasks that lead to a functional increase of the final product, providing a point of control and monitoring of the product to develop. This is a way to detect early problems throughout the project would tend to worsen.

2. Theoretical Perspective

In this software's development of we must take in consideration that concepts are not always well applied concepts and/or interpreted, in particular the concepts of process, methodology and life cycle.

The term "software development process" [1] is used to designate a sequence of actions, grouped into phases and tasks, executed in a systematic and standardized way, which are performed by actors with well-defined responsibilities, and that from a set of inputs produce a set of outputs.

This process involves four points of major importance:

- Guide the sequence of execution of the involved activities.
- Specify the descriptive models of the system to be developed.
- Manage the tasks of both participants and team as a whole.
- Provide criteria for monitoring, evaluation of models and project activities.

Thus, the methodology [1], as well as being a sequence of steps and recommended procedures to be applied during the process of developing information systems, also it requires the use of a set of tools, techniques and notations.

It also includes principle and rules that materialize in practice what is the theoretical concept of process, namely:

- Rules for preparation of estimates (cost, time).
- Techniques for taking measurements.
- Procedures to be followed to ensure the quality.
- Training programs.
- Models of documentation to produce, commonly called templates.
- Practical examples detailed.
- Techniques for the setting of the methodology, which can be applied to enable them to adapt to specific situations.

The concept of life cycle is, say, a synonymous expression of the concept of process, but reported to traditional approaches.

2.1. Waterfall Model vs Agile Software Development

The traditional waterfall methodology is currently one of the software development methodologies most widely used and is based on a control of attitude, in carrying out projects in a phased manner, well-defined and justified steps for an accurate prediction of the outcome.

There are several methodologies, but the best known is the waterfall model [2] that is characterized by the project having clearly defined levels of evolution, so there must be a clear idea of what the customer needs.

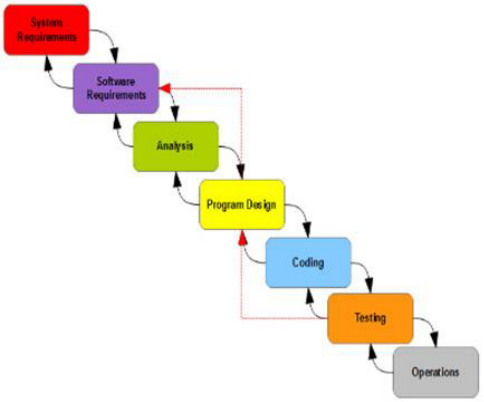

Fig. 1. Winston Royce's Waterfall Model.

Upon reaching a certain level, you get the result that is needed to be able to proceed to the next level , so you should be aware that the successes of the initial levels are of utmost importance for the success of the project, and that the cost of an error increases exponentially if detected in the final phase of the project , since it will involve redoing all of the previous phases, so, being an expensive and time consuming process, besides requiring reduced implementation times and a great capacity for change.

These methodologies generate enormous failure rates, being the factors for this failure:

- The requirements are not fully understood before beginning the project.
- Often the client knows just what they want after seeing an early version of the software.
- Requirements change frequently during the development process.
- New tools and technologies make the use of planning unpredictable.

These problems can be solved by a model in which all phases occur simultaneously: requirements; analysis; design; coding and testing. Thus, the simplest way would be to apply this model to build and deliver the application from beginning to end, and that all information would reside only in one person.

From another approach, this view could be applied to two developers through "peer programming" from the paradigm Extreme Programming [3]. However, this solution is still not scalable. Its necessary to apply this vision to larger teams, achieving at the same time that each element has a global overview of the product every day.

To solve this problem, a set of methodologies emerged called "agile" methodologies that are less focused on the overall control of the project and less bureaucratic, preparing the team for changes and problems that arise during the project development.

These methodologies have a distinct method of creating value for the customer: all steps are performed in short periods, offering the customer a frequent feedback on the work done, and allowing the change of priorities for the work according to what will bring you more value. In addition to the management of priorities, it is provided the

possibility of removing tasks initially planned introducing other ones, allowing the adjustment of the product in development to your vision and to the changes that the business requires.

However they should not be used for all problems, since the choice of the appropriate methodology for each project should be made in the concrete case attending the particularities of each case, namely the type, size, and risk.

In some cases, the best solution may be to adopt an intermediate method, and peculiarities of each other.

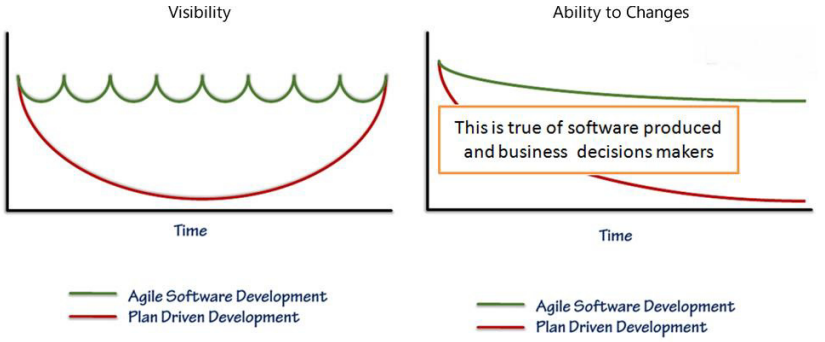

Fig. 2. Pan Driven Development vs Agile Software Development [6].

3. Scrum

Scrum is an agile development methodology which recognizes as natural the existence of changes and unexpected problems during the process of software development. Its main feature is the focus on the value for customer, always looking to first perform the tasks that undertake more value.

The Scrum methodology is structured in PDCA (Plan- Do- Check Act) cycles. In each of these cycles it is intended that a set of tasks to be performed leading to a functional increase of the final product, providing a point of control and monitoring of the product to develop. This is a way to detect early problems that throughout the project would tend to worsen.

Another characteristic of Scrum is that there is little or no documentation, the latter has been deprecated in favor of increased transparency implemented through increased customer interaction. Creating documentation for the completion of a project is a lengthy and expensive process, and not always correctly express the needs of both parties. On the one hand the development team needs much time to express correctly all requirements in documents, on the other hand, the owner of the product does not always have the necessary availability and/or competence to appreciate and interpret all documentation produced as well as to assess if it correctly specifies the desired product. Scrum encourages customer involvement in the design, minimizing documentation in favor of performing software that implements the interpretation of the team of the requirements set out by the client and the subsequent adjustment of the same. In certain environments (specifically in highly regulated environments such as health or listed companies) the existence of documentation cannot be undermined. In these cases the documentation itself is one of the characteristics of the product and should be seen as such, specifically its production should be prioritized according to the value that the product brings. The Scrum development cycle is illustrated in Figure 2 and starts with the prioritization of the features to be developed (Product Backlog).

At the beginning of each Sprint a planning meeting (Sprint Planning Meeting) is performed, where the features developed during the Sprint Product Backlog are taken and implemented in Sprint Backlog.

This meeting is typically held between the Product Owner and Scrum Master. The order in which the features will be withdrawn respect the priorities set by the Product Owner. Each functionality is partitioned into a set of tasks that joins the Sprint Backlog.

Being defined tasks to perform begins the Sprint during which the Team will develop the functionality. This may take two to four weeks, depending on the set on the Sprint Planning Meeting, and simultaneously covers the following parts of development: analysis; design; coding and testing.

Every day its performed a meeting (Sprint Daily Meeting) with an extremely short duration (typically not more than 15 minutes) for all elements to expose the situation in which their work is, what was developed since the last meeting, what is going to be done next , and if there are impediments to their progress. Also daily, the team proceeds to update the Sprint Backlog, introducing the hours spent on each task and updating the overall estimate for the completion of the task. Through these records and the number of total hours estimated for the completion of all tasks, is generated Burn Down Chart [4]. This serves as an indicator of the current state of the Sprint, indicating the progress regarding the estimates.

At the end of the Sprint it is performed another meeting (review) with the Product Owner where implemented features are demonstrated, thereby giving rise to an increment of the final product. After the team conducts a retrospective (Scrum Sprint Retrospective) to identify issues that have arisen during the Sprint, solving them and avoiding them in the next Sprint. Once completed the retrospective, it starts again the process proceeding to the planning of the next Sprint. This cycle will continue until all the features have been developed or until the product owner decides that the not yet developed features do not add enough value to justify the investment in construction. The process described here can also be extended to more than one team, each of which performs the same process independently and in parallel. The intersection point of each team will be a new Daily Meeting but this time performed only between the Scrum Masters.

Here, each of the Scrum Masters indicates the status of the development of his team, trying to solve together any kind of impediments that may arise to the progress of one of the teams. During the project development was adopted an updated version of the SSW (www.ssw.com.au) Scrum (Summer 2013).

4. Case Study

Given the fact that the development team as well as key partners, may not hold meetings and / or development of the face-to-face design, we adopted the tool that seemed more appropriate to our needs in (shared chat room, management system and creating feedback, version control, and others): Microsoft Visual Team Foundation Server 2013. Using the Team Foundation we got the controlling and neutralize all our problems adjacent to the fact that we have a virtual team where proximity was one of the problems to be addressed.

4.1. What is Microsoft Visual Team Foundation Server 2013?

The Microsoft Visual Studio Team Foundation Server 2013 allows the entire team to work more effectively collaborate and be more agile, while accumulating and sharing knowledge. This tool generates reports and dashboards that provide historical trending, full traceability and real-time visibility on the quality and progress against project objectives [4].

- **Better management of projects** - Facilitates project management by connecting teams of developers, testers, operations and other teams involved throughout the development cycles of software.
- **Easier collaboration between teams** - Service Management Lifecycle Applications to integrate projects and teams, consistently across the enterprise.
- **Visibility of the quality** - Business agility to support the full visibility of investments in software development, from early development to delivery.

4.2. Backlog Construction

This step is the construction of the backlog created in each Sprint Planning Meeting and lists the tasks to be performed so that the features are implemented. These tasks are inferred from features that were selected from the Product Backlog to accomplish during the Sprint in question.

Each task must have an estimated duration not exceeding 24 working hours (3 days). A task that exceeds this value should be divided into subtasks. Once development teams to self-organize, tasks present in this record are not awarded to any team member must each select the tasks you want to accomplish.

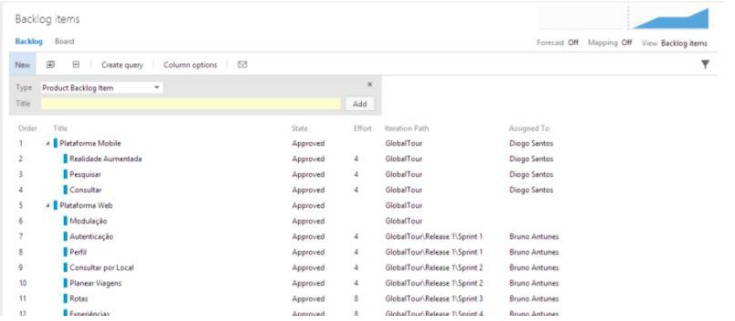

Fig. 3. Backlog Team Foundation Server.

4.3. Sprint Planning

This section is responsible for the meeting held at the beginning of each Sprint, where the team represented by the Scrum Master, the Product Owner negotiates with what features to withdraw from the Product Backlog to build the Sprint Backlog. After the selected objectives for the Sprint, the team meets and divides the functionality into smaller tasks to perform.

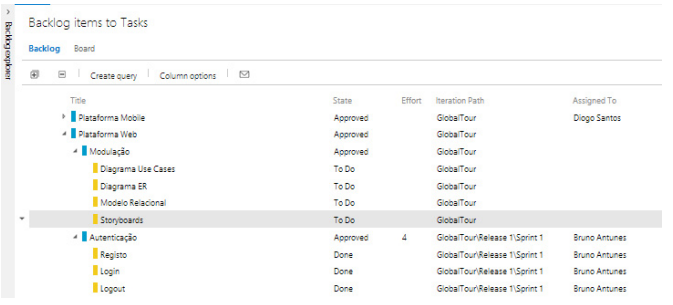

Fig. 4. List all the tasks of each Product Backlog Item.

4.4. Sprint

This is the period in which a set of features from the Product Backlog are developed. This period typically lasts between two and four weeks. The Figure 5 shows the status of number one sprint along the project development, and the transition phase, "To do" for "in progress" until you get to do "done".

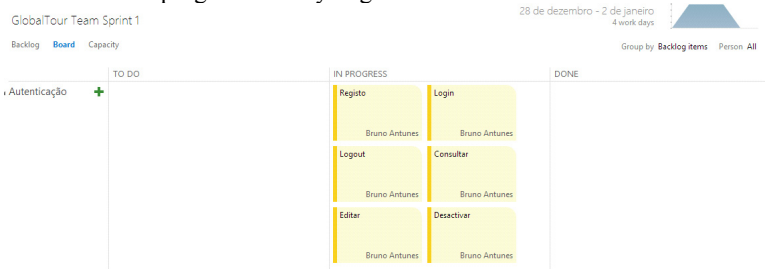

Fig. 5. Sprint 1.

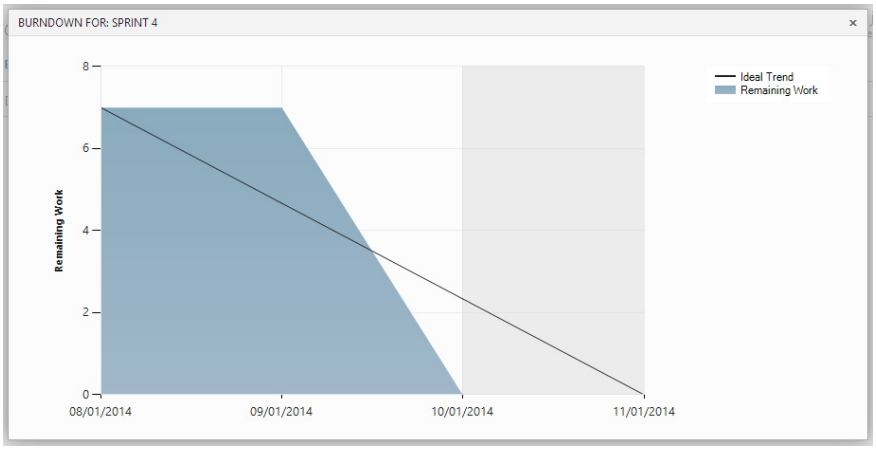

Fig. 6. Burn down Chart Sprint 4.

This is the period in which a set of features from the Product Backlog are developed. This period typically lasts between two and four weeks. The Figure 5 shows the status of number one sprint along the project development, and the transition phase, "To do" for "in progress" until you get to do "done".

The Burn down chart representing in Figure 6 have a little detour, because as the work was being developed, it was concluded that the tasks would not take the time planned development by not showing such complicated tasks as planned. We have also *Sprint Review* which occurs at the end of each sprint in order to demonstrate the work. This Allows the Product Owner to have an increasingly real vision of the end product.

4.5. Feedback Product Increment

In this section the feedback tool that Microsoft Visual Team Foundation Server provides feedback to the product will be displayed.

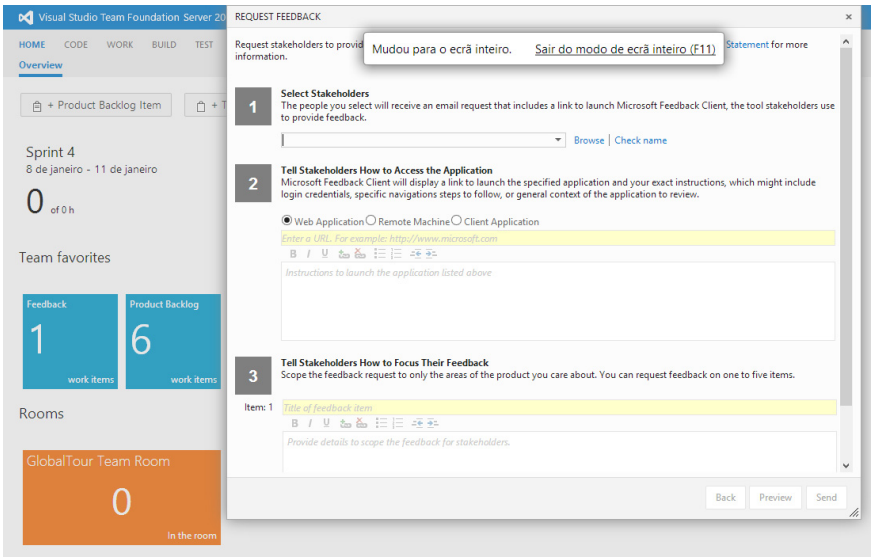

Fig. 7. Provide Feedback.

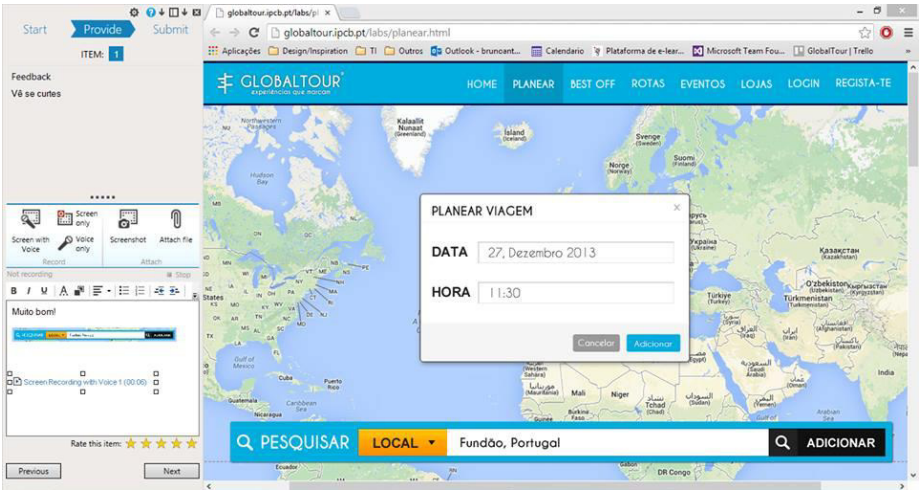

Fig. 8. Request Product Feedback.

Once the request for feedback is made, an email is sent to the product owner or a beta tester with the information needed to program the feedback Microsoft Visual Team Foundation Server 2013 installation.

5. Conclusion

After an analysis (Waterfall vs Agile) methodologies, we used an agile methodology: Scrum. To put this methodology into practice resorted to the use of Microsoft Visual Team Foundation Server 2013.

Given the fact that the development team as well as key partners, may not hold meetings and / or development of the face-to-face project, Microsoft Visual Team Foundation Server 2013 has proven to be a powerful tool and prepared to answer all challenges and limitations that face the virtual teams during software development.

Were then obtained closer to key partners through the feedback tool provided by Team Foundation and consequent development of the project came increasingly to the needs of the same, this time increasing the quality of the final product. In technical terms, the development of this application provided the contact with quite a large set of technologies requiring a great learning ability and autonomy, it is necessary to pay special attention to the integration between all of them.

This assessment allowed us to identify, in advance, a controlled problems that could delay the development of the application if they were found in the implementation phase environment.

References

[1] A. S. e. C. Videira, UML, Metodologias e Ferramentas Use Case, Portugal: Edições Centro Atlântico, 2014.

[2] W. W. Royce, Managing the Development of Large Software Systems: Concepts and Techniques, Los Angeles, USA: Technical Papers of Western Electronic Show and Convention (WesCon), August, 1Fig. 70.

[3] K. Beck, Extreme Programming Explained: Embrace Change. The XP Series. s.l., Addision Wesley, 1999.

[4] "Microsoft Technet," 17 Janeiro 2014. [Online]. Available: http://social.technet.microsoft.com/wiki/pt-br/contents/articles/19693.team-foundation-server-2013-visao-geral-e-conceitos-sobre-a-ferramenta.aspx.

[5] M. James, "Scrum Reference Card," 17 Janeiro 2014. [Online]. Available: http://scrumreferencecard.com/scrum-reference-card/.

[6] M. Habib, "Agile software development methodologies and how to apply them," Code Project, 2013 December 2013. [Online]. Available: http://www.codeproject.com/Articles/604417/Agile-software-development-methodologies-and-how-t. [Acedido em 30 March 2014].