An Agile Object-Oriented Method to Develop Modern Software Applications

Jose A. Gallud

***Abstract*—Agile methods and tools are being widely adopted in many software companies. The irruption of the Agile Manifesto has induced a revolution in the field of Software Engineering with the apparition of many agile methodologies. Among the most known agile methods are Scrum and eXtreme Programming (XP). This article describes Xcrum, a new Object-Oriented and agile method that is a combination of Scrum and XP that allows software teams to develop modern applications. Xcrum allows software teams not only to organize themselves in an agile manner, but also it allows them to develop applications following both Object-Oriented and agile principles. The article describes Xcrum and provides practical information, by means of an example, on how to develop a project using Xcrum.**

***Index Terms*—Agile, Scrum, XP, Object-Oriented.**

I. INTRODUCTION

THE field of Software Engineering has solved its nth growth crisis with the emergence of agile methodologies. These methodologies seem to free the heavy burden development teams were suffering while were applying documentation-based methodologies.

On the other hand, the Object Oriented programming paradigm is experiencing a second youth thanks to the new programming paradigms introduced by the Internet languages. Languages such as JavaScript, Python or Ruby are presented in many modern Web projects, and they are the appropriate technologies in projects where Xcrum is applied.

This article introduces Xcrum, a new agile proposal for the object-oriented development of software applications and services. The article describes the new proposal by describing its principles and the elements that comprises it. The final discussion analyzes the advantages of Xcrum.

The article is organized in the following sections. Section II presents a brief overview of the current landscape of software development. In the next section (III) the problem that you want to solve with this proposal is described. Section IV presents the principles on which Xcrum is based. Section V presents a general discussion of the proposal. Section VI describes what it is to develop an object-based solution. Section VII makes a proposal of Xcrum support tools. Section VIII contains an example of an Xcrum application, with special emphasis on obtaining the object-oriented solution. Finally, Section IX presents the conclusions and future work

J. A. Gallud was with the Escuela Superior de Ingeniera Informtica, Campus universitario s/n, 02001 Albacete (Spain),
E-mail: jose.gallud@uclm.es
Manuscript received February 2, 2018; revised .

II. RECENT ADVANCES IN SOFTWARE DEVELOPMENT AND AGILE METHODOLOGIES

One of the effects of this latest software crisis is the birth of agile methodologies. The appearance of the now famous Agile Manifesto [1] is the starting point of this new era. In its genesis is a group of software engineers who decided to rebel in the ”plan and document” style omnipresent in most development teams. At the time of the emergence of the Agile Manifesto, methodologies such as the cascade model, the Rational unified process (Rational Unified Process) [2], or Rapid Application Development (RAD) [3] dominated the development teams.

As noted by Fox and Patterson [4], the crisis of the 1960s led engineers to try to develop methodologies that made it possible to develop quality software and predictable and controlled budget. The results of this effort were a series of development processes based on planning and documenting (Plan-and-Document). Some strategies (process models) on which many of these development processes are based are well known: Cascade model, Spiral model and Iterative and Incremental models. In this last group, the RUP process should be located, which is a combination of the previous models. Development processes based on Planning-Document are tedious since they put a lot of emphasis on the preparation of documents, such as templates, memories and diagrams, which sometimes diverts attention from the software product that is intended to be developed.

The agile proposal tries to recover the lost prominence for the software product in favor of documents and diagrams. Alternatively, the Agile Manifesto also gives importance to the motivation of the development team, and to maintain fluid contact with the client.

As a result, agile methodologies began to emerge that tried to apply the principles listed in the Agile Manifesto. Among the best known we can mention Scrum [5] and Extreme Programming (better known as XP) [6].

From Scrum and XP, there are numerous agile proposals that have been emerging, although these are general proposals that are based on applying the principles contained in the agile manifesto. For example, DSM [7] is based on a principle in the integrates several statements of the Agile Manifesto: the best business value emerges when projects are aligned with clear business objectives, often visible results are released and involve motivated people. In this sense, Xcrum also applies the principles of the Agile Manifesto, as it is an agile method based on XP and Scrum.

Letelier, in [13], presents a catalog (AgileRoadmap) to implement agile practices in development teams, without suggesting a particular agile method. The Xcrum proposal is similar to AgileRoadmap in that it promotes agile practices, although Xcrum is based on principles that come from Scrum and XP, to which the object-oriented vision of the solution joins.

In [14], the authors summarize the best-known agile methods to point out the little attention paid to software architecture. The authors then propose a methodology to define software architecture in agile environments. In Xcrum, the software architecture is an essential element, since it is based on starting defining the solution as an object-oriented solution.

To conclude this section, Xcrum is related to most agile methods insofar as it applies the principles of the Agile Manifesto. However, it is novel in that it takes as reference two of them, Scrum and XP, to propose a synthesis of both.

III. The inception of Xcrum

The definition of a new method must be accompanied by a justification. This section presents the reasons that have motivated the appearance of Xcrum. This explanation can also be seen as the inception of Xcrum.

The review of agile proposals made in the previous section allows us to classify them into two conceptually wide groups:

- Those methodologies that are geared towards the agile management of the project, or even pose only principles to better manage a company. In this group we can include Scrum, Kanban and Lean [11]. We could tag these methedologies as management-oriented.
- Those methodologies that focus on applying an agile approach in obtaining the software product. In this group we include XP. These methodologies could be tagged as development-oriented methodologies.

The management-oriented methodologies are, as its tag suggests, heavily influenced by the dynamics of management. Scrum promotes close contact between the people in charge of the development team and the client's representative, promotes the early delivery of value, and establishes two levels to define the requirements (with the language of the client and the team's language), for example.

These methodologies, however, often do not provide guidelines or principles that help the development team in its task of obtaining software products, beyond the general recommendations.

Another disadvantage of the management-oriented methodologies is that, in an attempt to respond to all aspects of project management, they are no longer agile and become "heavy" proposals or closer to the methodologies of the Planning-and-Document style.

On the other hand, the development-oriented methodologies have the strong point of telling the team how they should work to achieve a quality product. For example, in XP it is indicated by the activities of Exploration of the Game of Planning, how to proceed to establish the content of the next sprint.

The methodologies oriented to the development team usually contain many recommendations and may become inflexible. For example, XP reaches the level of detailing how the space in which the team works should be organized. In XP, the programming by pairs is promoted, something that is not feasible in many situations.

In relation to software development, the object-oriented approach is taken as a guide for the design and coding of the software product.

Therefore, it would be desirable to have an agile method that combines the management aspects of the project that Scrum provides, with the aspects that guide the team in the development of the software product that XP provides. If this method also maintains the principles of object-oriented development, the software products obtained will have the advantages of this type of solution (flexible, extensible, modular, etc.).

IV. Foundations of Xcrum

Being an agile method, Xcrum tries to apply most of the principles of the Agile Manifesto. In particular, Xcrum emphasizes the following principles of the manifesto: prioritizing the early delivery of value, accepting that the requirements change, involving the customer, prioritizing the tested and functioning software and valuing simplicity. Being a proposal inspired by two existing ones, its novelty lies in the way it combines them.

Xcrum is based on three principles:

- Apply the Scrum approach in terms of software project management
- Apply the XP approach in terms of the development team and obtaining the software product
- Use the Object Oriented approach to software development

The following sections explain in more detail the principles on which Xcrum is based.

A. Apply Scrum in terms of project management

Xcrum assumes Scrum in terms of project management. This means that it employs the same skeleton, artifacts and meetings:

- Skeleton: the skeleton of Xcrum is the sprint, which lasts from 1 to 4 weeks
- Roles: Scrum and XP roles are similar, so any of them can be adopted. Scrum explains the existence of a team leader (Scrum Master)
- Artifacts: Scrum artifacts such as the Product Backlog, Sprint Backlog, Sprint Burndown and the most important artifact are assumed, namely, the code tested and functioning
- Meetings: Scrum meetings are also held as the Sprint Preparation Meeting, the Daily Meetings (Daily Scrum), and the Sprint Review Meeting (review and retrospective)

B. Apply elements of XP regarding the development team

Xcrum assumes some of the ideas of XP regarding the management of the development team and the obtaining of the software product.

Xcrum assumes the basic activities of development:

- Encode: is the activity par excellence
- Test: this activity gives value to the tests
- Listen: it is an activity that helps to understand what is asked
- Design: a good design is the key to obtain a good product

These activities are carried out through the Game of Planning with its three phases: Exploration, Commitment and Direction; and through the Planning of the Iteration, with its three phases: Exploration, Commitment and Direction. Each phase defines a series of activities.

C. Design an object-oriented solution

In Xcrum any approach to the solution must be carried out as a proposal based on a set of well-organized objects (design patterns) that interact with each other.

The object-oriented approach is present in XP implicitly. In Xcrum it is done explicitly. This means that in the approach of the solution it is recommended to begin by designing an object model. It is discouraged, in general, to start with persistence or the user interface. This programming model has the advantage of producing flexible and modular solutions.

V. Elements of Xcrum

This section defines the elements of Xcrum: roles, iteration, artifacts and meetings.

Like any good agile method, the most important artifact in Xcrum is the code tested and working. All other elements are means to get the software product.

A. Roles of Xcrum

The roles of Xcrum take into account the separation of responsibilities, business and techniques, suggested by the Agile Manifesto. There is separation and also complementarity since it is about that both work together in obtaining the solution.

Thus, Xcrum uses Business and Development as terms to define the roles that identify two responsibilities. Business defines the user stories and the priority of them. Development is responsible for estimating stories and converting them into code.

The Business role of Xcrum is equivalent to the Scrum Product Owner. The Development role is equivalent to the Scrum Team role. The ScrumMaster of Scrum is the leader of the development team at Xcrum.

B. Iteration in Xcrum

The heart of Xcrum is the Iteration in the same sense that the Sprint is for Scrum. The Iteration in Xcrum lasts from 1 to 4 weeks. At the end of the iteration, the team must provide an increase in value in the form of code tested and functioning.

The Xcrum Iteration is equivalent to the Scrum Sprint.

C. Xcrum artifacts

User stories: Xcrum is based on defining the system requirements (functional, non-functional and information) as user stories, in a similar way to other agile methodologies.

- List of System Histories: is the list of user stories defined by Business to describe a system. This list is equivalent to the Scrum Product Backlog.
- List of Iteration Stories: is the subset of stories that are assigned to an iteration. This list is equivalent to the Sprint Backlog.
- Progress Chart: is the diagram that in Scrum is called Sprint Burndown and that serves to measure the progress of the iteration (sprint).

As you can see, the Xcrum artifacts are basically those of Scrum.

D. Xcrum meetings

The Xcrum meetings follow the structure of Scrum, with incorporation of some XP activities. The following sections detail how XP activities are combined in Scrum meetings.

1) Iteration Preparation Meeting: The two roles, Business and Development, participate in this meeting. In Scrum the Sprint Preparation Meeting has two parts, in the first one it is carried out between Business and Development and its objective is to choose the functionality of the next Sprint. In Xcrum this meeting has the same objective, although it is proposed to incorporate the activities of the Exploration and Commitment phases of XP, namely:

- Write a story: Business writes a functionality
- Estimate a story: Development estimates the time
- Divide a story: If you can not estimate
- Sort the stories: Business order by value and Development by risk
- Choose field: Business chooses the end date of the Iteration or functionality (and Development date)

As you can see, this part of the preparation meeting of the Iteration takes advantage of the detail of activities that XP provides, while, in Scrum, it is left undefined.

The second part of the meeting corresponds to the team and consists of detailing the tasks in which each story is broken down.

2) Xcrum Daily Meeting: On a daily basis, the team reviews the status of the project following the same scheme proposed by the Daily Scrum. At this point some XP activities are proposed to obtain the increment:

- Accept a task: a developer chooses a pending task
- Implement a task: define the test cases, implement the task and integrate the code
- Recovery and reestimation: these are activities to readjust the load, or the dates, with respect to the estimate.

These XP activities, incorporated into Xcrum, serve to give content to the daily task of the team. The only difference in relation to XP is that in Xcrum it is not necessary to implement the task through the technique of programming in pairs.

An important aspect of Xcrum are the tests. The team must write test cases so that the meaning of ”finished” is demanding, not thinking about Business, but internally, thinking about the team.

In the day to day of development the third principle of Xcrum is put into play: Object Oriented Solution. We will deal with this aspect in a later section (section VI).

3) Iteration Review Meeting: In Scrum, at the end of the Sprint, two meetings are held: the Review meeting and the Retrospective meeting. The first is done with the Product Owner. The second is internal to the team.

In Xcrum, a similar scheme to Scrum is proposed. In the first meeting (Review) the increase in value of this Business Iteration is shown to check if it is what you requested. The second part is that Development review the realization of

VI. The Object-Oriented approach in Xcrum

Since the appearance of the book Design Patterns [8], the paradigm of Object Oriented programming has experienced a growing development that continues to influence many of the software solutions regardless of the technology used.

In recent years we have witnessed, in Web programming, a renaissance of dynamic languages, some of which have appeared for some time, such as Javascript, Python, Ruby and PHP. While most dynamic languages can be said to be object oriented and class oriented. However, languages such as Javascript (at least until version 7), omit the definition of classes, since they are languages oriented to prototypes and functions, although it is possible to use them with the same approach used in the object-oriented and class-oriented paradigm.

There are many advantages to using the object-oriented and class-oriented paradigm, as opposed to prototype-oriented. However, this discussion is outside the scope of this article.

What interests us here is to define how to obtain an object-oriented solution, and how it is integrated into the Xcrum methodology.

Obtaining an object-oriented solution is based on a single principle:

The object model comes first

This principle establishes as a priority to first define the object model of our application. This means that both the user interface and persistence must be left for later. In this way, and in broad strokes, the proposal for the content of the iterations follows this sequence:

- Examine functionality based on user stories
- Define the object model (with the tests)
- Define the service layer
- Define the user interface
- Define the persistence layer
- Final deployment

This principle applies more easily to Web-based solutions. In these environments, the first phase (define the object model), can be detailed as follows:

- Develop the model objects in the client: using the tools provided by most browsers we can validate our object model
- Define the tests using some testing framework
- Move the model to the server: the ideal solution is for the client's own object solution to be the one we use on the server. This depends on the technology chosen
- Adapt the tests to the model on the server

The first step (develop the object model) is also broken down into activities that are repeated until we get the complete object model. Here we apply an XP principle by which we design (and implement) only what is needed, avoiding including everything we know in advance that we will need. So, the object model is obtained with the following steps:

1) Choose a functionality (from the list of user stories)
2) Draw the class diagram with the minimum that is needed to implement that functionality
3) The team conducts a discussion of the model
4) Implement the classes following the diagram
5) Validation in the browser
6) Optional tests are implemented: It is not convenient to write the tests too early to avoid rewriting. This contradicts an XP principle but is more practical.
7) Go back to step 1

The previous steps are completed when the developed model contemplates a sufficient set of functions.

This procedure is especially useful when using the same technology for client and server (such as Javascript and NodeJS). The benefit of using the same technology is applied to the tests, since the same set of test cases, with minimal changes, is valid on the server side.

In a later section a practical example is presented.

VII. Tools used in Xcrum

As for the Xcrum tools, any of the tools that support Scrum can be used. In particular, it is recommended to use Kunagi [9]. Kunagi is a tool that supports all Scrum artifacts and, therefore, can also be used to track Xcrum.

Kunagi has the added advantage of including some artifacts from other methodologies, such as the task artifact (pending, assigned and done) of Kanban [12].

In a secondary level, since we are talking about agile methodologies, it is suggested to use tools for the modeling of UML artifacts such as StarUML. These tools are useful for the development team because they allow them to specify in a diagram (classes, sequence) a concept of the solution that allows to discuss it with the team, using that diagram, and adopt a solution.

The development team discusses different ideas, often expressed in a diagram, and adopts some of the proposed solutions. These tools are especially important for object-oriented solutions.

It is convenient to have a tool for version control and repository of the solution and other artifacts.

VIII. Case study: Xcrum in action

Xcrum has been applied, although without using that name, during the last three years in the academic field. Specifically, the students of the subject Software Engineering Processes of the School of Computer Engineering had the task of carrying out a complete project of a solution based on REST services, from the initial idea to its actual deployment in suppliers such as Heroku [10].

In this section we will see how Xcrum was applied to the development of one of the projects proposed in the aforementioned subject. The project consisted in developing The Goose game based on Internet and multiplayer. The game should allow two or more players to play The Goose from their browser.

As the course consists of 15 weeks, short iterations of 1 week were established. Sometimes some iterations had to be extended, taking 2 weeks.

The iterations were marked, in general lines by the stages that have been presented in section VI:

- Define the set of functions of the game of The Goose: this phase is simple since it is a board game with squares, several players and a die.
- Define the object model
- Define the user interface
- Define persistence: this step was not applied in this case

The work teams were formed by the students (Development role), while the teacher exercised the role of Business, although sometimes he also participated as team leader.

At the end of the class the iteration was planned, establishing the delivery. At the beginning of the class the revision tasks were carried out

The stage ”define the object model” actually comprises a set of steps, as explained in section VI. Let's see an example of how they would apply to the case of the Goose project:

1) Choose a functionality: suppose we choose to start by making an initial approach to the game: ”The game has a board”.
2) Draw the class diagram with the minimum needed to implement that functionality: in our case we would start with the Game and Board entities associated with a relationship.
3) The team conducts a discussion of the model: The team can discuss attributes and responsibilities. At this moment they realize that a board (entity Board) is composed of a set of boxes (entity Box).
4) Implement the classes following the diagram: the implementation of these classes is quite direct. Let's see the code that should be generated from the class diagram (in Javascript):

```
function Game(board){
  this.board = board;
}
function Board(){
    this.boxes = [];
}
function Box(){
}
```

5) Validation in the browser: this step allows us to obtain a first validation of our object solution using the browser console. For this we have to edit our code in a Javascript file that we must include in an index.html file. This index.html file is the one that we invoke from the browser. We use the browser console to create our instances. Let's see an example of the creation of a Game instance:

```
game = new Game();
```

6) Optional tests are implemented: The tests could focus on checking that the game has a board and that the board consists of boxes. However, it is preferable to start the tests when we have something more complete model.
7) Go back to step 1.

From this mini iteration to design and develop the object model, the solution grows almost alone. Here are numerous issues that give rise to specific code:

- How do we add squares to the board ?: this results in a method in the board type ”addCase ()”
- How do we identify a box ?: The answer to this question allows us to define the attributes ”position” and ”subject” in Box.
- What else does the game have ?: the Goose game needs chips, which requires a new entity, Ficha, in the model.

This way of proceeding is a concrete application of the agile principle of designing only what is needed at that moment.

In addition, the browser console allows us to test our object model as it provides us with a dynamic execution environment with which we can create instances (and even modify them).

It would take a lot of pages to illustrate the complete example but the process to follow is pretty well defined. By way of example, Fig. 1 shows the simplified class diagram of iteration 4.

For more information you can use the author's GitHub repository (https://github.com/jgallud/laOca) where there is a more advanced version of the project.

IX. Conclusion

This article presents Xcrum, an agile and object-oriented method for the development of applications. Xcrum has been inspired by Scrum and eXtreme Programming. The article presents an example of an Xcrum application that illustrates in some detail how the principle of designing an object-oriented solution is carried out. In summary, Xcrum can be beneficial in those projects where you want to apply agile principles with an object-oriented approach to the solution.

References

[1] Agile Manifesto. Retrieved from: http://www.agilemanifesto.org (2001).
[2] P. Kruchten. The Rational Unified Process: An Introduction. Addison-Wesley Professional (2003).
[3] J. Martin. Rapid Application Development. MacMillan Publishing Co. Ed (1990).
[4] A. Fox and D. Patterson. Engineering Software as a Service. An Agile Approach Using Cloud Computing. Strawberry Canyon LLC (2014).
[5] K. Schwaber. Agile Project Management with Scrum. Microsoft Press (2004).
[6] K. Beck. Extreme Programming Explained. Addison-Wesley; 2nd edition (2004)
[7] The DSM Agile Project Framework. Retrieved from: https://www.agilebusiness.org/resources/dsdm-handbooks/the-dsdm-agile-project-framework-2014-onwards
[8] E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (2011)
[9] Kunagi. Retrieved from: http://kunagi.org/
[10] Heroku. Retrieved from: https://www.heroku.com/
[11] M. Poppiendieck, T. Poppiendieck. Lean Software Development. Addison-Wesley (2003).
[12] R. Louis. Custom Kanban: Designing the System to Meet the Needs of Your Environment. University Park, IL: Productivity Press. (2006)

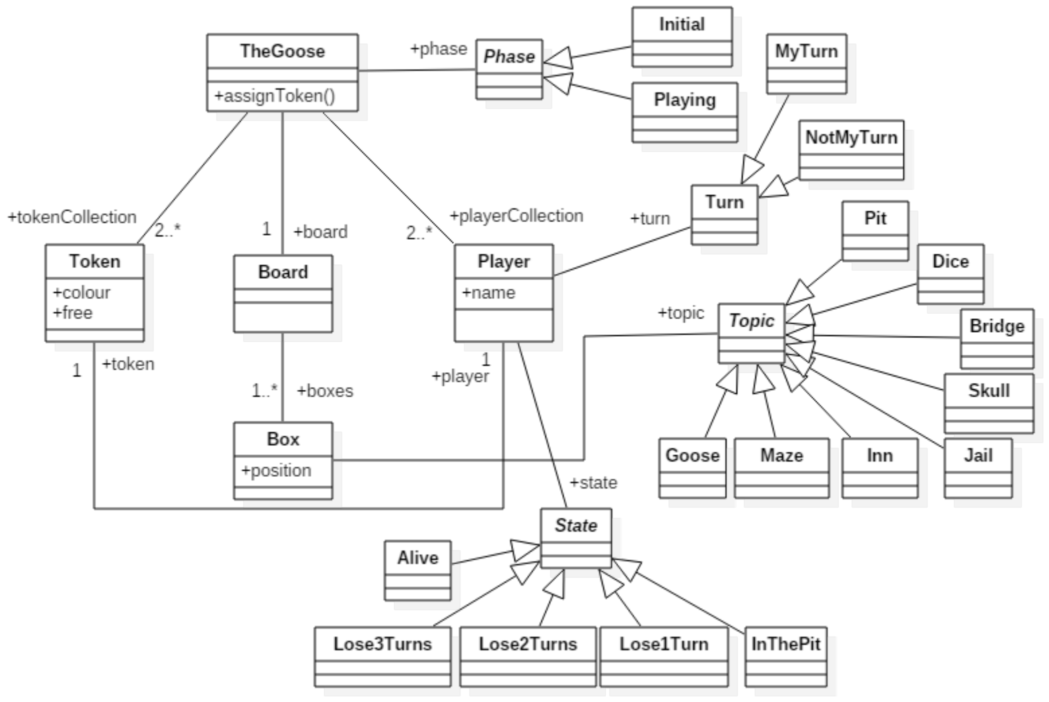

Fig. 1. Class diagram of Iteration 4

[13] P. Letelier, M. C. Penads. AgileRoadmap: An Approach to Implement Agile Practices in Teams. IEEE Latin America Transactions, VOL. 15, NO. 7, (2017)
[14] M. Mekni, G. Mounika, C. Sandeep, B. Gayathri. Software Architecture Methodology in Agile Environments. J Inform Tech Softw Eng 7: 195. doi: 10.4172/2165-7866.1000195 (2017)