

Tool Support for the XP Game

Frank Keenan, Adrian Parke, Namgyal Damdul

Department of Computing and Mathematics, Dundalk Institute of Technology, Ireland
{frank.keenan, namgyal.damdul}@dkit.ie.

Abstract. Engaging and motivating students is always a challenge particularly when teaching subjects that do not have a high practical content. The XP Game is a simple and interactive set of activities that simulates the eXtreme Programming (XP) development process. However, the activities used are not computing specific and the game requires a substantial group of people to realize the intended benefits. This paper reports on the development of prototype tool support for the XP Game with the expectation that it could be used with undergraduate Computing students.

Keywords: XP Game, eXtreme Programming.

1 Introduction

Agile Methods (AMs) provide an alternative to traditional approaches to software development by promoting individuals and interactions, working software, customer collaboration and the ability to respond to change. XP has been successfully delivered as part of many software engineering courses. Examples include [2][3][4][5]. However, it does need to be carefully adapted for each situation [6][7]. The XP Game [8] is widely used in industry and academia to introduce aspects of the XP process.

The XP Game is a simple game, lasting between two and three hours, that is often used in industry to introduce the XP process. It involves groups of individuals planning and carrying out simple tasks in competition with other groups. However, despite the many benefits reported for the XP Game it still requires a substantial group of people to be available at the same time and the tasks to complete are general. The proposal here is to develop a tool to automate the XP Game, using software development specific examples, so that individuals or small groups of software engineering students can use it to understand the XP approach to planning. Section 2 presents an overview of XP, with the XP Game explained in section 3. An overview of the prototype tool is presented in section 4.

2 eXtreme Programming

eXtreme Programming (XP) emphasises communication, flexibility, regular releases of working software and close collaboration with customers. XP allows software organisations to get fully-functioning software to the customer faster and with increased customer satisfaction. An onsite customer is responsible for all 'business problems', leaving developers free to focus on technical decisions [1]. The role of the customer centres on the development of the system's requirements and starts during the *Planning Game*. These are written in non-technical terms on cards as *user stories* each of which describes the requirement, its associated acceptance test, and an estimate of the time and effort needed to implement the requirement. Next, stories are sorted and prioritised, and allocated to a development iteration. In the subsequent *Release Planning* phase, a decision is taken on which stories to include in each release. Each release is divided into a number of short (1 – 4 weeks) iterations, each of which will complete a number of stories. A key additional strength of agile processes is their ability to cope with changing requirements, allowing new stories to be added and existing stories altered or removed from proposed iterations.

To increase the likelihood of the system meeting the sponsor's requirements, AMs advocate that the customer should be co-located with the development team throughout the project. In this way, the customer is available to advise on development decisions, and ensure that delays on key development issues are avoided.

The next section presents an overview of the XP Game.

3 The XP Game

The XP Game [8] is a simple and highly interactive game that simulates the XP development process. (The instructions and props are freely available at <http://www.xp.be/xpgame.html>). It is used in industry to introduce development teams and their customers to XP. No technical knowledge or skill is required to participate. An element of competition helps the players bond and remain focused. By the end of the game, everybody has experienced how user stories, estimation, planning, implementation, and functional tests are used.

The players are divided into teams of between four and eight. A coach helps each team explaining the rules and clarifying each aspect. For each task completed the team collects business points with the team accumulating the most winning. The team is given a set of prewritten story cards and the props required for the first iteration. These cards describe simple tasks, such as "Build a two-story house of cards," "Blow Up Ten Balloons," or "Find a missing card from a pack of cards".

The team members, acting as *developers*, estimate the time required to implement each task. Values between 10 and 60 seconds, or *impossible*, are chosen. The next objective is to consider the *customer* perspective. It is necessary to create a release plan by prioritizing the stories to implement to achieve the highest amount of business points. The total implementation time is 3 minutes. Once an activity begins the clock starts and only stops when that activity is achieved or cancelled. To achieve the

business points for an activity it must pass the acceptance test validated by the coach. For example, the acceptance test specified for the task “Blow Up Ten Balloons” might have been that each should be of 30 cm diameter. The clock restarts when the next activity starts. At the end of the iteration the points are counted and there is time for discussion. Concepts introduced during the game can also be explained. The second and third iterations proceed in a similar manner.

4 Proposed Solution

The proposed solution is to create an automated version of the XP Game. The potential benefit is that students or staff can learn the basics of the XP development process wherever or whenever they want to. The game itself will be a web-based game. The main programming languages that have been used are Java, HTML, SQL and CSS. The player’s information and the game data are stored in a Database system (Microsoft Access). A user can create a profile, edit their profile and delete their profile, perform an *iteration* and perform a practice Iteration. An Administrator can create, add or modify stories.

The game is played one iteration at a time. The player can see each story name, a one line description and business value. For each, they must indicate a perceived complexity and use this in conjunction with the business to derive a prioritised iteration plan. They ‘execute’ the stories in the order of the plan. When a story is to be executed the detail of that story is displayed. For example, the particular story represented in Figure 1 is to *find 2 errors in 12 lines of code*. The user enters the number of the lines that has the syntax error. The player moves to the next story, having accumulated the business value for successful completion. Other sample stories relate to analysis (*identify the class that receives message X*) and implementation (*Add one line of code to do Y*).

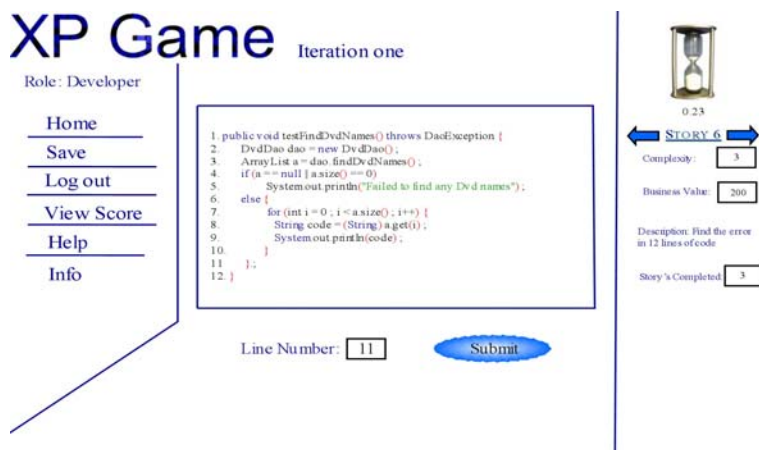


Figure 1: Game Interface with Sample Story

The user interface with a sample story is displayed in Figure 1. Generally, for each story the *complexity* level and *business value* are displayed. Also, the *elapsed time* for the iteration and *Stories Completed* to date is indicated. A summary of progress is presented in the *iteration score card* as outlined in Figure 2. On completion the total accumulated business value and the project velocity, measuring accuracy of estimation, is displayed.

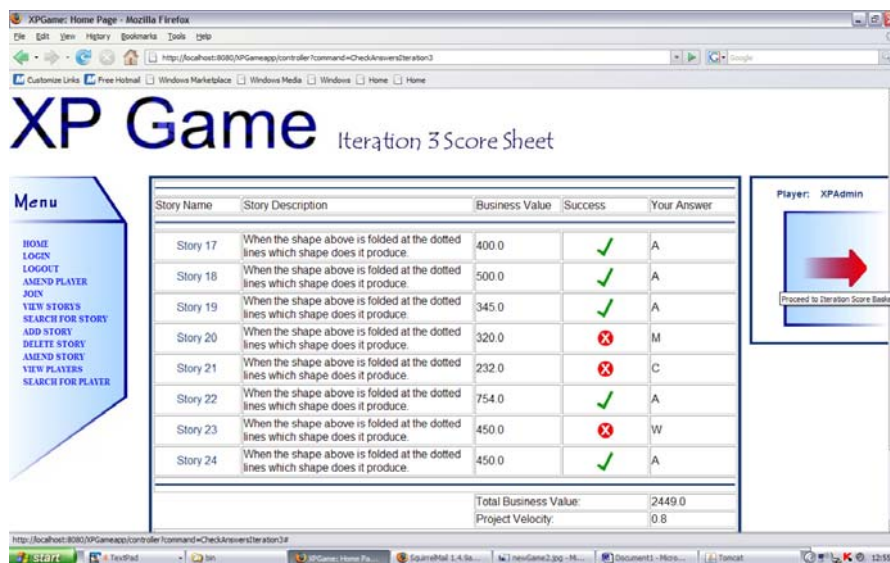


Figure 2: Iteration Score Card

The prototype has been evaluated by final year Honours candidates in Commercial Computing at Dundalk Institute of Technology. Each student was given a list of tasks to complete. During this, they were observed, while on completion they were interviewed. The general response was positive with the feedback showing that the automated game was *enjoyable, easy to play*, and importantly helped to improve their *understanding* of the *Planning Game* practice. The participants had previous experience of performing this XP practice and they felt that automated game demonstrated the process very well. The responses indicated that the *countdown clock* accurately reflected the *time* constraint of a real project and that this made the game feel competitive even though there was no physical competition. The summary iteration screen was *clear* and *easy to read*. This helped to highlight areas that need improvement or warranted more time. Also, new skills were learned.

The feedback collected from these observations and interviews also helped to identify problems requiring improvement. General problems that were encountered were in *navigation* and *on screen help*. Also, particular functional errors were detected.

3 Conclusion

The XP Game is used to help participants understand the concepts of the XP Planning Game in a practical and fun way. However, no tool exists to allow individuals or groups to benefit in an educational setting. The proposed solution presents a prototype that automates the XP Game allowing participation without physical engagement. Another advantage of this tool is that the result can be obtained without taking much time. Such a tool can be augmented with additional features to easily understand other difficult features of XP. For example, it could be used to help students understand XP's pair programming concept of Driver-Observer roles.

References

- [1] Beck, K., *Extreme Programming Explained: Embrace Change*, 2e, Addison-Wesley, Boston, 2005.
- [2] Hedin G., Bendix L., Magnusson B. "Introducing Software Engineering by means of Extreme Programming", 25th International Conference on Software Engineering 2003.
- [3] LeJeune N., "Teaching Software Engineering practices with Extreme Programming", *Journal of Computing Sciences in Colleges*, February 2006, Volume 21 Issue 3
- [4] Shukla A., Williams L., "Adapting Extreme Programming For A Core Software Engineering Course", *Proceedings of the 15th Conference on Software Engineering Education and Training*, Kentucky, 2002.
- [5] Flohr T., Schneider T., "An XP Experiment with Students – Setup and Problems", *The 6th International Conference on Product Focused Software Process Improvement*, Oulu, Finland, June, 2005
- [6] Schneider J, Johnston L. "eXtreme Programming – Helpful or Harmful in Educating Undergraduates?", *Journal of Systems and Software*, vol. 17, Issue 2, January 2005.
- [7] Noble J., Marshall S., Marshall Stephen, Biddle R. "Less Extreme Programming", *Australasian Computing Education Conference (ACE) 2004*, Dunedin, New Zealand.
- [8] Peeters V., Van Cauwenberghe P., "The XP Game Explained", Chapter 27, *Extreme Programming Perspectives*, 2003.
- [9] Pascal Cauwenberghe. *XP Game Description*, Available: <http://www.xp.be/xpgame/description.html> [6th March, 2008].