Teaching TDD, the Coding Dojo Style

Ramiro Batista da Luz*, Adolfo Gustavo Serra Seca Neto† and Robinson Vida Noronha‡
*Systems development department
City Chamber of Curitiba
Barão do Rio Branco St., S/N, Downtown, Curitiba, Paraná, Brazil.
Email: ramiroluz@gmail.com
†Academic Department of Informatics
Federal University of Technology - Paraná
Curitiba, Paraná
Email: adolfo@utfpr.edu.br
‡Academic Department of Electronics
Federal University of Technology - Paraná
Curitiba, Paraná
Email: vida@utfpr.edu.br

***Abstract*—Coding Dojo is a dynamic and collaborative activity inspired in martial arts that follows a discipline in a cheerful and pleasant environment. Several agile development techniques are used during the Coding Dojo, such as test driven development, pair programming and baby steps. This article presents some Coding Dojo characteristics that help teaching agile development techniques. Some experienced practitioners were interviewed to get qualitative information about their perception of the Coding Dojo practice. An electronic survey was answered by Coding Dojo participants from several Coding Dojo groups in Brazil, which provided quantitative data. This article presents Coding Dojo as a teaching technique to help developers create software with higher test coverage rates.**

***Keywords*-test driven development; teaching; agile; Coding Dojo.**

I. Introduction

The word *Dojo*, translated from the Japanese, means “place of the way”. It was used originally to designate the meditation space of Buddhist monks. In this context, it can be translated as “the place to study life”. Nowadays it is well known as the place where martial arts are practiced.

Coding *Dojo* is an activity that has been used by programmers to practice agile practices, specially test driven development (TDD) [1], pair programming [2] and baby steps [1]. In a Coding Dojo session, programmers get together to learn with each other in a friendly and social environment.

Coding Dojo was created by Laurent Bossavit and Emmanuel Gaillot [3] inspired by the idea of Coding Katas, created by Dave Thomas [4].

We are searching for evidences of Coding Dojo advantages or disadvantages as a tool for teaching test driven development, when compared to traditional classes. By traditional classes, we mean theory presentation in lectures with exercises to be solved individually by the students.

A. How does it work?

To run a Coding Dojo session, the first step is to choose a programming challenge to be used in the session as a way to practice programming. There is no need to solve or finish the challenge during the session. In time-limited turns, one pair works on the solution, using one computer, doing small steps in the direction of the solution, using test driven development and, most important, communicating the idea to the audience. After the time box ends, the pilot, which is the developer that is coding, goes back to the audience. The copilot, which is the developer helping the pilot, starts coding. And one volunteer from the audience becomes the copilot. In Figure 1 we show a picture of a Coding Dojo session in a university conducted by the first author. It is possible to see the audience, the pilot-copilot pair and the computer screen projection.

The activity allows the participants to be of any level. Each turn can have one expert and one beginner in the pair, so they can learn together.

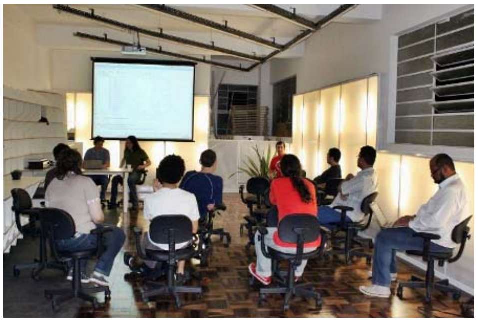

Figure 1. Coding Dojo Session

1) Agile practices: Test Driven Development, shortened as TDD, is a set of techniques that “encourages simple

978-0-7695-5009-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ICALT.2013.114

designs and test suites that inspire confidence" [5]. There are two basic rules: write new code only if an automated test has failed and eliminate duplication. These two rules imply an order to the tasks of programming, known as the TDD mantra *red-green-refactor*. *Red* - first write a simple test that fails; *green* - make the test work quickly; and *refactor* - eliminate any duplication created in merely getting the test to work [5].

Pair programming is a dialog between two people simultaneously programming. This practice intends to keep each programmer on task, brainstorming refinements to the system, clarifying ideas, taking initiative when their partner is stuck and holding each other accountable to the team's practices [1].

It is always tempting to make big changes in big steps. The "Baby steps" practice acknowledges that the overhead of small steps is much less than when a team wastefully recoils from aborted big changes [1]. This practice is particularly used in TDD, which proceeds one test at a time [1].

Coding Dojo has been used by programmers to learn and to train agile practices, specially test driven development [1], pair programming [2] and baby steps [1].

A Coding Dojo session needs one projector, one computer, one white board and comfortable space for the practitioners. It is not necessary but very appreciated to have some food after the session, giving the participants a moment to socialize and relax. In this environment, good conversation and even some personal projects may start.

II. Objective

The objective of this research is to verify the Coding Dojo activity as a way to teach Test Driven Development to obtain software with better test coverage rates.

III. State of the art

Studies related to agile methods adoption and agile methods in education are relatively common. In [6], for instance, a TDD training was created and gave some evidence that hands-on exercises can be good for teaching TDD.

A comparative case study on the impact of test driven development compared three projects measuring the effect of test driven development on software design using object-oriented metrics, showing that the effect of test driven development on program design was not so evident as expected, but showing that the test coverage was significantly superior to iterative test-last development [7].

A set of experiments were ran to empirically assess different parameters of the test driven development in a university environment comparing test driven development to a iterative test-last development process. Preliminary results showed that test driven development is not substantially different from test-last development process [8].

A report shared the experience of running a Coding Dojo at the University of São Paulo, Brazil and at a private company. The role of a Coding Dojo in the learning process was discussed, showing how students at different skill levels can use deliberate practice to improve and share knowledge with a wider group [9].

According to [10], agile practices such as pair programming or test driven development are best learned when they are actually performed rather than read about. Mariana Bravo proposed two methods that might reinforce the learning of some Agile practices. The first is Coding Dojo and the second is the use of tools to give automated feedback while the learner works.

IV. Methodology

Coding Dojo is an activity that involves people and that have many subjective aspects so it has to be studied considering as many details as possible. To achieve this the authors realized some Coding Dojo sessions and got the feedback of the participants who reported what they liked and what they disliked in the session. Those feedbacks helped to create two surveys, one was answered by graduation students in person during a Coding Dojo session, the other was sent to Coding Dojo groups in their discussion lists. Then, the authors interviewed some experienced Coding Dojo organizers.

A. Students survey

This survey was used as a prototype to know who was attending to the Coding Dojo sessions at the university, answered by ten students. Some questions were used to discover the preferences of the students regarding the preferred frequency of the Coding Dojo sessions and their preferred programming language.

B. Electronic Survey

The objective of the electronic survey was to check the participants perception of the benefits of Coding Dojo as a technique to learn agile practices, such as Test Driven Development, Pair Programming and Baby Steps.

The electronic survey (available at http://va.mu/XcTM) was released and answered by Coding Dojo practitioners from several regions in Brazil. Using the Likert scale, the survey asked the opinion about the Coding Dojo practice and its effects in the learning process of agile practices.

C. Interviews

The teaching and learning process is a human relationship which is not simple to understand and it is not easy to measure using only quantitative data. To better understand the subjective aspects of the Coding Dojo technique, semi-structured interviews were used to guide the interviewer and at the same time to allow the respondent to express his opinion. As said by Steinar Kvale, with qualitative research interviews you try to understand something from the subjects point of view and to uncover the meaning of their experiences [11]. The interviews were recorded using a video camera.

V. Results

The results were collected from the surveys and semi-structured interviews. The first survey was just a pilot study. This first survey just helped to formulate the questions for the on-line survey, as the participants had not much experience with Coding Dojos. The authors published one article [12] in the Brazilian Symposium of Informatics on Education. Some of the results are presented as charts in the Surveys V-A subsection.

A. Surveys

The on-line survey got 64 answers. Figure 2 shows the distribution of the participants regarding the experience in Coding Dojo sessions.

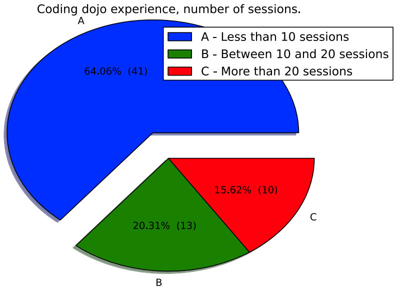

Figure 2. Coding Dojo number of sessions

The programming experience in years of the surveys respondents are shown in Figure 3.

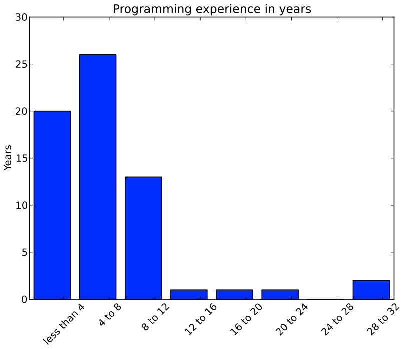

Figure 3. Programming experience in years

The majority of the volunteers agreed that baby steps help to solve challenges gradually. Only 17% disagreed or are neutral to the affirmation which can be seen in Figure 4.

The participants believe that writing tests before coding helps the progress of the activity. In Figure 5, we see that 77% agree or strongly agree with this affirmation.

The participants agree that pair programming helps the leveling of the attendees (Figure 6).

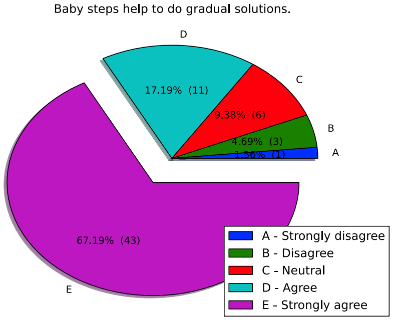

Figure 4. Baby steps Coding Dojo practitioner's perception.

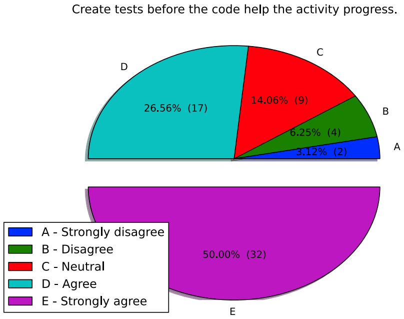

Figure 5. Test driven development Coding Dojo practitioner's perception.

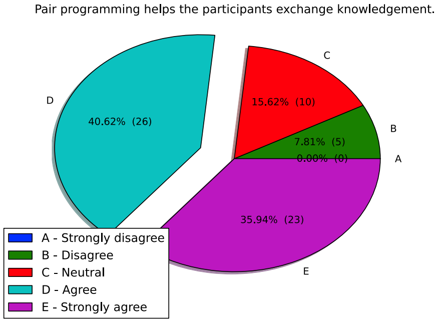

Figure 6. Pair programming Coding Dojo practitioner's perception.

B. Interviews

The authors recorded interviews with some experienced Coding Dojo organizers.

1) Elizabeth Leddy: Elizabeth Leddy is a software consultant, member of the Plone[1] core developers team and work as a volunteer at a hacker space, named Noisebridge, in San Francisco [13], California. There, she teaches web programming with the Python programming language [14], HTML and Javascript to people that want to change their

[1] A free software content management system.

careers.

She reported that the Coding Dojo helped to keep a reasonable pace, depending on the pace of the student that was using the computer, also the students started interacting more.

2) Alexandre Freire: Alexandre Freire is a master in Computer Science from the University of São Paulo (USP). According to him the advantages of the Coding Dojo are the repetition, the motivation and the programming languages diversity, participation and communication, is good to practical activities. One disadvantage is that it is focused on practice, so it is not good for theory classes.

3) Daniel Wildt: Daniel Wildt, a respected Brazilian agilista, has been working with software development for 15 years and worked as university teacher for 7 years. He said that, after the adoption of Coding Dojo, the students started to show more interest in the class and the participation grew considerably.

4) Daniel Cukier: Daniel Cukier is technology director at, a marketplace for handcraft professionals. His first experience with Coding Dojo was at Mathematics and Statistics Institute of São Paulo University - IME-USP, with Danilo Sato and Hugo Corbucci. He said that Coding Dojo helped Locaweb employees to understand what is TDD, pair programming and the participants noticed that Coding Dojo is useful to socialize the development and improve the networking. He reported that the Coding Dojo was used as a tool for recruiting at Locaweb. At the end he said that the Coding Dojo stimulate the team work and interaction.

5) Danilo Sato: Danilo Sato is consultant at Thoughtworks. When he was at the XP 2007 conference he participate of a Coding Dojo Workshop organized by Emilie Bachè, Backing to Brazil he invited some friends from the Mathematical and Statistic Institute of the São Paulo University - IME-USP to try the novelty. Danilo believes that Coding Dojo is more interactive than lecture classes. He said that Coding Dojo favors the teaching of pair programming and TDD.

6) Hugo Corbucci: Hugo Corbucci is a programmer and consultant, actually he works at Thoughtworks. His first experience with Coding Dojo was with Danilo Sato when he organized the first Coding Dojo meeting at the Mathematical and Statistic Institute of the São Paulo University - IME-USP. He co-authored one paper accepted at agile.com in 2008 [9]. According to Hugo the Coding Dojo offers a collaborative environment that favors the participation on the other side, Coding Dojo isn't good for situations that the student need to be alone and silent to reflect about a topic. He cited some agile practices most indicated to teach with Coding Dojo as pair programming, unit tests, TDD after the learning of unit tests, and incremental development.

7) Mauricio Aniche: Mauricio Aniche works as consultant and instructor at Caelum, a training company. His Coding Dojo experience started at the Mathematical and Statistics Institute of São Paulo University - IME-USP. According to Mauricio, Coding Dojo is a tool that improve the participants socialization. He said that Coding Dojo can't substitute lecture classes but it is good to be used combined with lectures. For him Coding Dojo is specially indicated to teach pair programming, TDD and design practices as well to teach how to use continuous integration tools enforcing the continuous integration discipline. He believe that Coding Dojo can be used to teach programming at universities or training centers.

8) Roberto Rodrigues: Roberto Rodrigues is graduating at the School of Arts, Science and Humanity of São Paulo University - EACH-USP, works on the Center of Competency on Free Software of São Paulo University - CCSL-USP. He is a researcher of usability and agile methodologies. Started with Coding Dojos at the Mathematics Institute of Technology of São Paulo University - IME-USP. He highlighted the collaborative aspect of Coding Dojo. He told that Coding Dojo favors the practical learning. Roberto told that one disadvantage of Coding Dojo is it's focus on practice, easily losing some concept, so it is necessary to do some effort to compensate this gap. The Coding Dojo is specially indicated to teach agile practices that require discipline of are repetitive such as TDD, continuous integration and review, said Roberto.

VI. Results analyses

The survey was answered by 64 participants and Figure 2 shows that the majority (64%) of the answers came from people that were in less than ten Coding Dojo sessions. Taking this into account, all the results here have to be interpreted with care. Regarding Baby Steps, the survey indicates that it is a practice that helps gradual solutions and simplifies the process of finding a solution to a problem, as summarized in Figure 4. The results of the survey show that the use of test driven development in a Coding Dojo session helps the progress of the activity and that the Coding Dojo helps to learn test driven development, as shown in Figure 5. Pair programming helps the leveling of the group and it is good for the Coding Dojo session (Figure 6). These results give us evidences that the Coding Dojo activity is a good way to teach practical agile techniques.

A. *Coding Dojo advantages to teach TDD*

Test driven development, as many programming activities, need to be exercised. It is better to learn it by practicing. Coding Dojo is an activity focused on practice as pointed out by the experts we interviewed. Some of its characteristics can enforce the learning of TDD. The participation of students make them be an integral part of the teaching experience. They have to code and explain their code and they review code by other students. Coding Dojo also has some characteristics of problem-based learning (PBL), which is an instructional method where relevant problems are introduced at the beginning of the instruction cycle and are used to provide context and motivation for the learning that follows [15]. These characteristics together make Coding Dojo an interactive way to teach TDD, changing the way people learn how to program.

VII. Conclusion

At this moment it is possible to notice that the Coding Dojo is a dynamic activity which favors the participation and collaboration in an inclusive learning environment where students are able to participate in the teaching process showed by papers like [16] and [9]. Coding Dojo is very focused on practice, so it is indicated to repetitive and disciplined tasks as TDD, pair programming and continuous integration.

Acknowledgment

The authors would like to thank Danilo Sato, Hugo Corbucci, Daniel Wildt, Daniel Cuckier, Alexandre Freire, Roberto Leite de Moraes Rodrigues, Mauricio Aniche and Elizabeth Leddy for the interviews and all the survey participants which helped this research.

rbl
December 29, 2012

References

[1] K. Beck and C. Andres, *Extreme programming explained: embrace change (2nd edition)*. Addison-Wesley Professional, 2004.

[2] L. Williams and R. Kessler, *Pair Programming Illuminated*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[3] L. Bossavit and E. Gaillot, "The coder's dojo - a different way to teach and learn programming," in *Extreme programming and agile processes in software engineering*, ser. Lecture Notes in Computer Science, H. Baumeister, M. Marchesi, and M. Holcombe, Eds. Springer Berlin / Heidelberg, 2005, vol. 3556, pp. 1156–1158, 10.1007/11499053_54. [Online]. Available: http://dx.doi.org/10.1007/11499053_54

[4] Dave Thomas, "CodeKata: code kata," http://codekata.pragprog.com/2007/01/code_kata_backg.html, 28 Jan. 2007. [Online]. Available: http://codekata.pragprog.com/2007/01/code_kata_backg.html

[5] K. Beck, *Test driven development: by example*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.

[6] B. Vodde and L. Koskela, "Learning test-driven development by counting lines," *IEEE Softw.*, vol. 24, pp. 74–79, 2007.

[7] M. Siniaalto and P. Abrahamsson, "A comparative case study on the impact of test-driven development on program design and test coverage," in *Proceedings of the first international symposium on empirical software engineering and measurement*, ser. ESEM '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 275–284. [Online]. Available: http://dx.doi.org/10.1109/ESEM.2007.2

[8] M. Pancur, M. Ciglaric, M. Trampus, and T. Vidmar, "Towards empirical evaluation of test-driven development in a university environment," in *The IEEE region 8 EUROCON 2003. computer as a tool.*, vol. 2. Ljubljana, Slovenia: IEEE, 2003, pp. 83–86. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1248153

[9] D. T. Sato, H. Corbucci, and M. V. Bravo, "Coding dojo: an environment for learning and sharing agile practices," *AGILE Conference*, vol. 0, pp. 459–464, 2008.

[10] M. Bravo and A. Goldman, "Reinforcing the learning of agile practices using coding Dojos," in *Agile processes in software engineering and extreme programming*, ser. Lecture Notes in Business Information Processing, A. Sillitti, A. Martin, X. Wang, E. Whitworth, W. Aalst, J. Mylopoulos, N. M. Sadeh, M. J. Shaw, and C. Szyperski, Eds. Springer Berlin Heidelberg, 2010, vol. 48, pp. 379–380, 10.1007/978-3-642-13054-0_41. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13054-0_41

[11] S. Kvale, *Interviews: an introduction to qualitative research interviewing*. Sage Publications, 1996. [Online]. Available: http://books.google.com.br/books?id=tJPZAAAAMAAJ

[12] R. Luz and A. Neto, "Usando Dojos de Programacao para o Ensino de Desenvolvimento Dirigido por Testes," *Anais do Simpósio Brasileiro de Informática na Educação. ISSN 2316-6533.*, Nov. 2012.

[13] E. Leddy, "San francisco - HackerspaceWiki," http://hackerspaces.org/wiki/San_Francisco, 2007. [Online]. Available: http://hackerspaces.org/wiki/San_Francisco

[14] ——, "PyClass - python classes using coding dojo," https://www.noisebridge.net/wiki/PyClass, 2009. [Online]. Available: https://www.noisebridge.net/wiki/PyClass

[15] M. Prince, "Does active learning work? A review of the research," *JOURNAL OF ENGINEERING EDUCATION*, vol. 93, no. 3, Jul. 2004. [Online]. Available: http://www.konferenslund.se/pp/TAPPS_Prince.pdf

[16] L. Bossavit and E. Gaillot, "The coder's dojo – a different way to teach and learn programming," in *Extreme programming and agile processes in software engineering*, ser. Lecture Notes in Computer Science, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, H. Baumeister, M. Marchesi, and M. Holcombe, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, vol. 3556, pp. 290–291. [Online]. Available: http://www.springerlink.com/index/10.1007/11499053_54