

TEACHING SOFTWARE PROJECT MANAGEMENT: THE ANARCHIC APPROACH

Adrian Firculescu^{1*}
Ion Bucur²

ABSTRACT

Throughout time, education has faced several adjustments depending on the nature of the subject it aimed to teach. Although anarchy has often been confused with chaos, an anarchical system tries to achieve social stability by encouraging individuals to make self-initiated decisions towards the benefit of the community. This paper aims to apply that anarchic philosophy in an academic-based environment, while teaching Software Project Management, and adapting these principles to a computer science study field.

KEYWORDS: *software project management, educational models, teaching strategies, collaborative learning, cooperative learning, competition-based learning, mixed collaborative-competition learning, independent learning, anarchy learning.*

INTRODUCTION

Among the different oriented approaches, the following can be distinguished as efficient learning processes: collaborative, competitive, anarchic, independent and mixed.

This document focuses on the concept of anarchic management. Anarchy is a term that is often misunderstood, so it should be clarified from the start. An anarchist seeks to abolish the state; thus the fundamental principle is that it holds no moral authority over the individual [1]. Therefore, anarchy is often confused with chaos – the lack of a form of authority – which is not what this ideology proposes.

Anarchism can be viewed as a stable society where the government fundamentals are initiated by the individuals. Given these principles, anarchy outlines a community of people based on reciprocity and equality.

As opposed to other learning processes, the anarchic learning approach abandons deeply rooted classical teaching rules and promotes the “survival of the fittest” psychology among students confronted to the method. This means that opportunities have to be found and gained [4], implying that power bases are earned, not given, and are based on knowledge, initiative and the capability to employ oneself. By using this method, students learn proactively how to search information, reflect and act. Thus, they become aware of the fact that success is achieved through one’s ability to create one’s own career path.

^{1*} corresponding author, Eng., “Politehnica” University of Bucharest, 060042 Bucharest, Romania, adrian.firculescu@cti.pub.ro

² Associate Professor PhD Eng., “Politehnica” University of Bucharest, 060042 Bucharest, Romania, ion.bucur@cs.pub.ro

However, anarchic approaches limit the learning possibilities due to unclear roles, tasks and lacking systematization, which may lead to improperly exploited personal resources, develop insecurity and encourage competitive-aggressive behavior.

PREVIOUS WORK

The anarchic approach is implemented in multinational companies such as Google, GitHub and Valve and is also a key element in education.

Within the university framework, the academic library reflects the characteristics of an organized anarchy. Academic libraries possess unclear organizational technologies. There is little overall agreement in the field as to what services a library should provide or how these services relate to the outputs of the educational process [5].

PROPOSED APPROACH

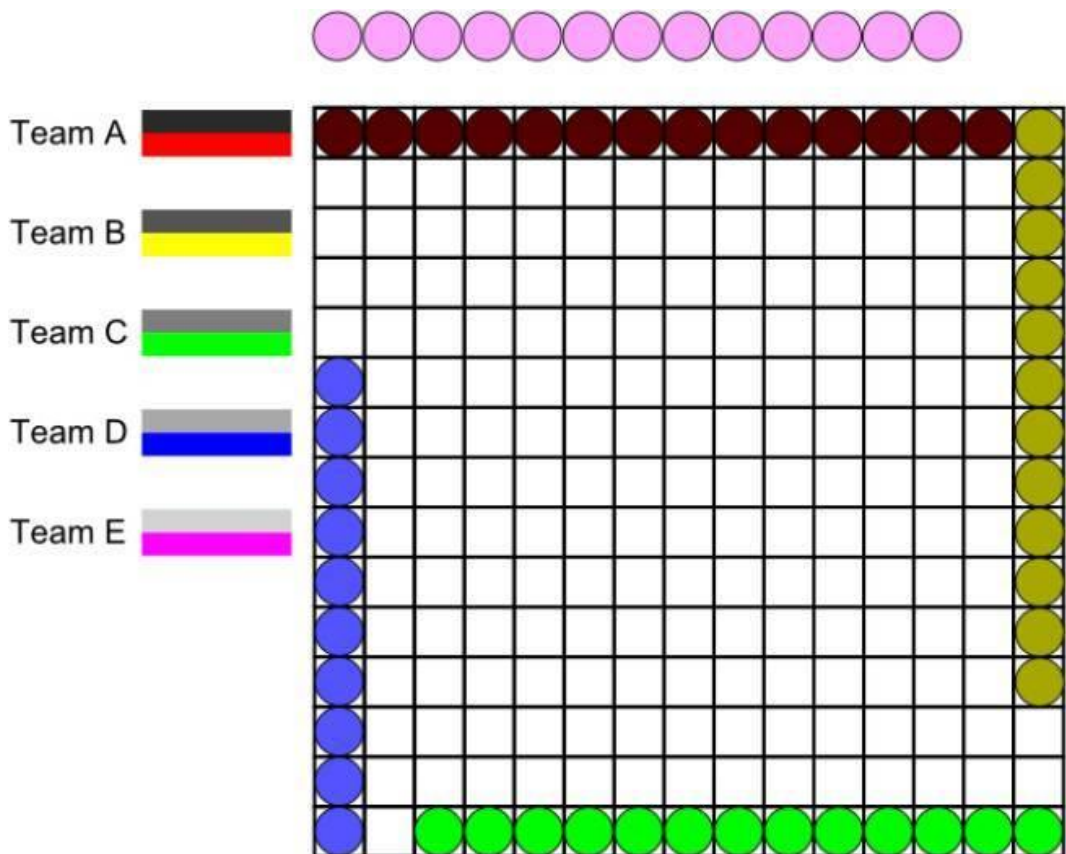


Figure 1. Start Game

During the Software Project Management course from the POLITEHNICA University of Bucharest, one particular project was proposed in order to illustrate the anarchic approach.

The project was actually a board game, in which each player (student) represented a pawn with a unique code. The purpose of each player was to ensure their own teams' success by protecting their team mates while creating a clear path so that one team member could reach the opposite side of the board.

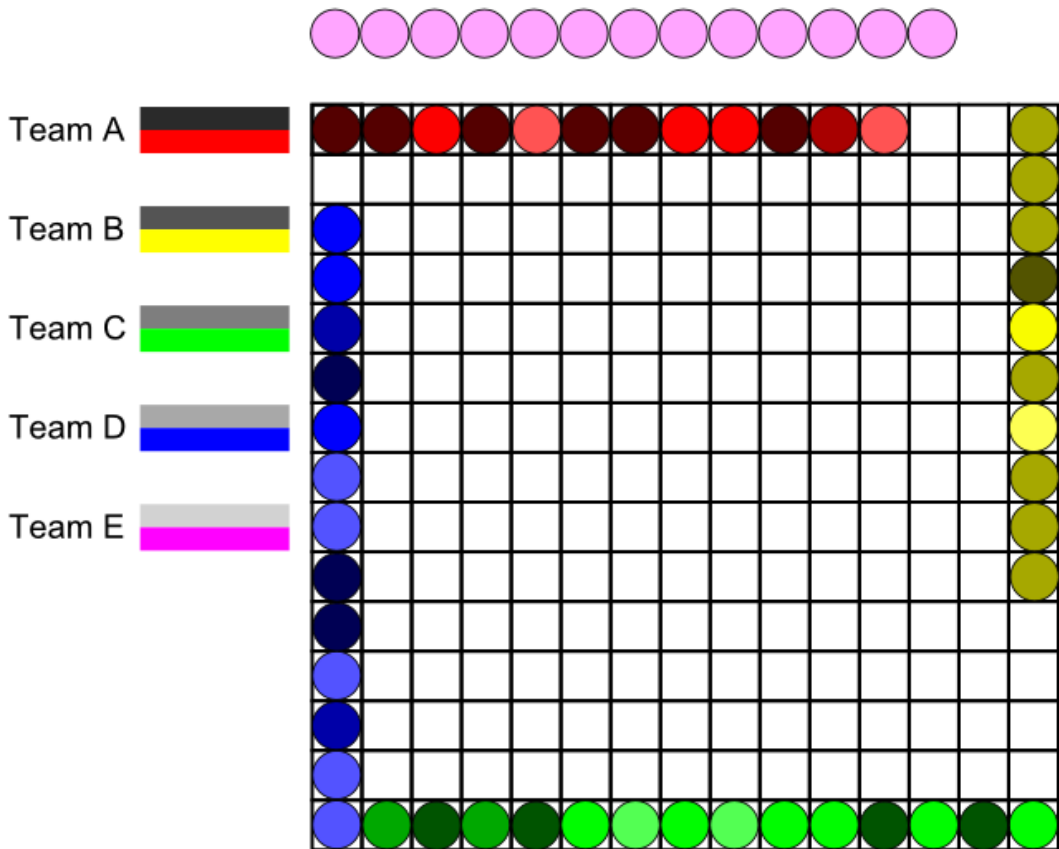


Figure 2. Shuffling – 5 teams playing

When a game starts, pawns are shuffled between teams, therefore one player may end up in a different team, while some would remain in the original formation. The number of pawns in a team sizes would be randomly determined. This mechanism of having opponent pawns within the team eliminates the possibility of implementing a predetermined strategy based on having a leader that would know where a pawn may end up on the board.

An important aspect of the game is that the pawn can reconsider its move at any moment in the round. For example, if a pawn is set to move forward, it could switch to moving in a different direction, therefore making an independent decision. Thus, a pawn may have an individual strategy, enhanced by algorithms or other heuristics. These would be helpful in identifying a saboteur pawn and analyzing its strategy in order to gain a surprise advantage. Moreover, a pawn may be programmed to act against any predetermined team

strategy, but it cannot know if the team it acts against is its original team, or it has been redistributed in a competing team.

Every time a team wins, all the members, whether redistributed or not, earn a point and the team, represented by the color, earns a point, too.

The individual grade for the application is calculated from the number of times a pawn belonged to a winning team and the group grade is calculated from the number of times the original team (color) won.

Since the selection process was anarchic, no specific rule for the role allocation among the participants was imposed. Therefore, flexibility was encouraged and opportunities arose for student socialization and positive interpersonal relationships.

In order for the anarchic approach to reach its goals, setting the subject of the project according to the students' knowledge is essential.

Students lack motivation when confronted with fields where they have little to no skills. Students should have some milestones set and receive periodical feedback on their work; otherwise there is a great chance that they lose their interest in the project.

The anarchic approach suggests that programmers work directly with the customer, which builds more trust and understanding about how the software development life cycle is affecting the final product.

The elements of the Organized Anarchy are described below [2]:

- I. Problems
- II. Solutions
- III. Participants
- IV. Flow structures:
 - Unrestricted / Democratic;
 - Hierarchical: Priority is given to important actors, problems, or solutions;
 - Specialized: Creating specialized problems and actions solved by nominated members.

Some well-known models illustrate the anarchic learning methodology. The first one is called Lindblom's muddling-through model. Charles Lindblom came up with the model by looking at decisions made in connection with public-sector budgets in the USA [3]. The advantages include the simplicity of the method and the efficient use of the time resource.

This method is characterized by realism because little changes can be handled with ease by the decision-makers. On the other hand, the overall picture of the project fades into the background, due to the lack of an in-depth debate.

The second anarchic model is known as the garbage can. The opportunity to make a decision is described in this model as a garbage can into which the organization's decision-makers put all kinds of different problems and solutions [3].

This model infers that the problems find their solutions – or fail to – following (anarchic) paths.

CONCLUSIONS

Since communication is one of the key aspects of the anarchic management, an educational environment is a very good ground to experiment this approach. Satisfactory results can be achieved because individuals know each other well; they have common goals, and similar philosophies within the domain in which they interact.

Also, ideas should originate from all directions, decisions should be made based on discussion and mutuality, and processes reflect respect for the individual even as they respond to the needs of the whole. These factors are fundamental for a successful social anarchy.

The anarchical approach on projects is a phase every student should have to face in their development. Because its usage has increased, especially in software developing companies, getting used to it may help students in their future careers.

The success of this type of management lays especially in the team composition. Some teams may be fit for this leadership style, while others may be not.

It is hard to impose this kind of freedom on students that are used to have others take decisions for them and that need guidance all through the project. It is even harder to do this with students that do not possess knowledge in the field of the project or lack motivation.

The perfect team for this kind of leadership is one in which members can collaborate with each other without supervision, including highly passionate and professional members.

Still, this kind of approach brings some advantages when using it for education. First of all, grades are a much better reflection on the student's submitted effort. In this case lazy students won't thrive on the work of others while hard working students will be rewarded.

Moreover, the anarchical management allows students to improve or develop skills such as: the ability to manage tasks and think on their own, the ability to come up with their own ideas. Nevertheless, it might teach them to isolate themselves from the group, losing the ability to collaborate with others.

FUTURE WORK

There are many ways in which the anarchic approach can be combined with other work methodologies.

A. Pair Programming

Pair programming should be taken into consideration, as it is a largely known collaborative learning method. Students program in pairs, not individually. This approach significantly improves students' programming competency, increases the likelihood of students getting a major degree in computer science and working in the same field in the future.

During pair programming, students work *two-by-two* on the same computer while completing regular programming assignments. The "*driver*" controls the mouse and keyboard while the "*navigator*" makes suggestions, points out errors, and asks questions.

The partners routinely switch roles to gain the benefits of each role. If there is a group of students working on a project, they could discuss ideas, make decisions and reach conclusions as a group, just like in the anarchic approach, but when the actual developing starts, they could choose a partner and do pair programming. Doing this, they take full advantage of both methodologies.

B. Jigsaw Procedure

The anarchic approach could also work well combined with cooperative learning. For example, the Jigsaw Procedure is a method that ensures the organization of classroom activities, so that it makes students dependent on each other to succeed. It divides classes into groups and decomposes assignments into pieces that the group assembles to complete the (jigsaw) puzzle.

The technique splits classes into mixed groups to work on small problems that are afterwards collated into a final outcome. The groups can then take an anarchic approach of solving the problem at hand by discussing and making decisions as a team, without relying on a leader.

C. Optimizing Programming Anarchy

A project should be viewed as a final product, regardless of the stage it is currently in. Given the fact that every individual can choose tasks on their own, important changes can have repercussions on the other team members.

An interesting concept is dividing the project into a *master* – a model designed for production and *individual branches* [1].

The main goal is to provide a master that is ready to use and has completed all the given features. The freedom of the individual is respected by the existence of the branches in which the individual could work without affecting everyone. The established rules should imply organized freedom. An individual can make changes to the master code only if the respective code has clarity, can function alone, and includes tests for all the code changes.

Another important aspect is communication. The work in progress that is divided into individual branches should be known by all group members and summarized [1]:

- Internal processes (individual's decision)
 1. Formulate an approach
 2. Write your code and tests
 3. Organize your branch
- External processes (interaction with the community)
 4. Present code for review
 5. Iterations based on feedback
 6. QA & deploy

In conclusion, given the amplitude of this learning process, this topic leaves plenty of room for further research and conduction of case studies. More details about alternate educational approaches in teaching Software Project Management can be found in [6-7].

ACKNOWLEDGEMENTS

The authors would like to thank teaching assistant Mihai Zaharescu and students Tudor Zugravu, Oana Spatarelu, Andrei Vasilescu, Codina Mihalache for their great support and assistance with this paper.

REFERENCES

- [1] Barun Singh, "Social Anarchy in Software Development". Original talk, December 11, 2012. Available Online: [http:// barunsingh.com/ 2012/ 12/ 13/ anarchy_in_software.html](http://barunsingh.com/2012/12/13/anarchy_in_software.html). Accessed at: February 5, 2016
- [2] Michael D. Cohen, James G. March, "Leadership and Ambiguity", Harvard Business School Press, 1986, pp. 3
- [3] Daniel A. McFarland. Organizational Analysis. Presented at online course on Coursera.org, October 2012, [https:// www.coursera.org/ course/ organanalysis](https://www.coursera.org/course/organanalysis). Accessed at: February 5, 2016
- [4] Henrik Holt Larsen, "Oticon: Unorthodox project-based management and careers in a "spaghetti organization"" HR. Human Resource Planning, Vol 25, No 4, 2002, pp. 30-37
- [5] Joan Giesecke, "Creativity and innovation in an Organized Anarchy", Journal of Library Administration, Vol 14, No 2, 1991, pp. 59-71. [http:// dx.doi.org/ 10.1300/J 111v14n02_05](http://dx.doi.org/10.1300/J111v14n02_05)
- [6] Costin-Anton Boiangiu, Alexandru Constantin, Diana Deliu, Alina Mirion, Adrian Firculescu, "Balancing Competition and Collaboration in a Mixed Learning Method", International Journal of Education and Information Technologies, ISSN: 2074-1316, Volume 10, 2016, pp. 73-81
- [7] Costin-Anton Boiangiu, Adrian Firculescu, Nicolae Cretu, "Combining Independence and Cooperation as One Anarchic-Style Learning Method", International Journal of Systems Applications, Engineering & Development, ISSN: 2074-1308, Volume 10, 2016, pp. 97-105.