



Use of Software Engineering techniques by independent game developers in Brazil

Bruno L. Carli

Unicesumar, Curitiba, PR, Brazil.

Email: brunolcarli@gmail.com

In this study, I investigated whether Brazilian independent (“indie”) game developers use methods and techniques derived from Software Engineering when developing their games. The hypotheses raised in this article are that, even with the vast literature available to guide good development practices, independent developers do not have specialists in their teams in the role of engineer; also, they do not use Software Engineering knowledge when developing their games. All this sums up to several difficulties during game development. Thirty-five indie developers from four Brazilian Internet communities and 13 Facebook groups were interviewed for this study, showing that indie game development in Brazil still lacks professionalism, especially regarding methodological aspects.

DELIMITATION OF THE SUBJECT

A definition of “indie” is given by Lemes (2009: 27 [my translation]): “*a project to be developed without the financial contributions of big companies (...) a game developed by a small team, or individually, by pure passion of the*

subject or simply to one day make money and start a career in the area of creation and development of digital games” (see Wikipedia, 2017b, for more information).

There is a bunch of indie games out in the market, some known far and wide, like *Minecraft* and *Angry Birds*, but some famous only within the gaming community, like *To the Moon* (Fig. 1) and *Stardew Valley*. In the Brazilian indie scene, *Chroma Squad* (Fig. 2) and *Momodora* (Fig. 3) are good examples of the latter case.

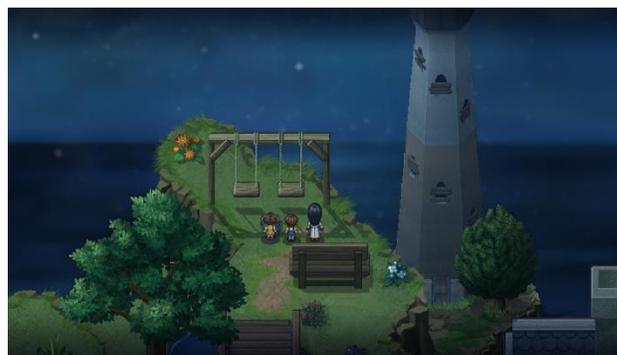


Figure 1. *To the Moon*. Screenshot of the game.

The process of game development is (or should be) bustling with engineering techniques, such as project organization, modeling, software metrics, surveying

requirements, and software documentation. According to Pressman (2010: 31), Software Engineering “encompasses a set of three fundamental elements – methods, tools and procedures – that enables the manager to control the software development process and provides the professional with a basis for building high quality software productively.” However, it is important to point out that Software Engineering was not always present in the development processes among professionals in the field. Pressman (2010: 8) states that at first “programming was seen as ‘an art form’. There were few formal methods and few people used them. The programmer often learned his trade through trial and error. The technical bragging and the challenges of building computer software have created a mystique that few managers cared to penetrate. The software world was virtually undisciplined.”



Figure 2. *Chroma Squad*. Screenshot of the game.

A digital game is by its nature a computer software and it must go through similar, although not identical, processes during their development. Velasquez (2009: 30 [my translation]) states that, contrary to the usual popular opinion, a “computer game is not just a

toy, but a large and complex software project developed by a vast team of professionals.” Therefore, similar problems can be detected during the development phases of games and “regular” software, such as: the long production time, the difficulty in measuring progress while the software is being developed, the lack of data collection during development, the late detection of errors, etc. (Pressman, 2010).



Figure 3. *Momodora: Reverie Under the Moonlight*. Screenshot of the game.

Moreover, even with similarities, game development differs in some instances from conventional software development (Morais & Silva, 2009) and still lacks a Software Engineering model dedicated to it (Velasquez, 2009). The importance of engineering methods in game development (and design) are even more obvious when thinking of the final game/software as a product for the market (Lemes, 2009; Lacerda & Selleri, 2012).

In summary, there is agreement in the literature that there is a need for engineering methods in game development, which should be different from conventional methods and

adapted to the specificities of games. By applying such methods, it is possible to maintain a stable project progress control, which will in turn result in a better product. As pointed out by Lacerda & Selleri (2012), the best candidate for this methodology lies in the area called Software Engineering.

The main roles on a team of game developers are: Programmer, Artist, Designer, Producer, Tester, Composer, Sound Designer and Editor (Doolwind, 2017; Wikipedia, 2017c). Among these, the Producer not only oversees the entire team, but is also responsible for the aspects of Software Engineering, including project management. The activities of the Producer are typically undertaken by software engineers (the titles assigned to these positions vary a lot: Engineer, Manager, Game Designer, etc.), evidencing the necessary presence of the engineer in a game development team. Without proper systematization during game development, even the best ideas will fail (Lemes, 2009).

This systematization must start before the production of the game, when the game design is defined and documented (Lemes, 2009): the GDD (Game Design Document) serves as the blueprint from which a game will be built, Sayenko (2015) has a great article describing how and why you have to write a good GDD; one of his tips for coming up with an effective GDD is to put just one person in control of it. From then on, it is the responsibility of the game designer (the engineer) to maintain the GDD.

It is clear that large game companies have specialized software engineers, but independent developers possibly do not. If

indie developers lack a person skilled in engineering techniques, they will likely neglect engineering aspects. As stated above, without giving proper importance to such aspects, even the best ideas will not save the project. Therefore, here I analyzed the reality of indie developers in my home country, Brazil. I investigated: (1) if they have specialists in their teams to fill the role of the engineer; (2) if they actually use techniques from Software Engineering for developing their games; and (3) if they have defined and followed a GDD. It was hypothesized that the indie community in Brazil do not comply with the three topics above.

METHODOLOGY

The first step of this study was a survey of the largest Brazilian independent game developer communities, which are mainly based on Internet forums and social media platforms. Only those groups on Facebook with 1,000 or more registered users and online forums with 1,000 or more registered users that are receptive (that is, accept the request to join the group in a period of seven days and do not exclude the search of the feed) were selected (Tables 1 and 2).

The second step was the preparation and application of a questionnaire (see the Appendix) to the groups and forums selected on the first step. The *Google Forms* platform was used for this, as it allows the preparation of online surveys. The questionnaire was semi-open, with objective questions of single and multiple choice, also counting with fields for (optional) further comments and explanations. The questionnaire was composed of 16 questions in total and was presented to the

groups outlined in Tables 1 and 2. The third step consisted in analyzing, interpreting and exposing the collected data in a statistical

manner. In this way, the initial hypotheses raised in this study was put to the test.

Table 1. Indie game developers groups on Facebook (last access: 03/Apr/2017).

Group	Members	Website
Construct 2 Br	2,946	https://www.facebook.com/groups/construct2brasil/
Construct 2 Brasil	4,584	https://www.facebook.com/groups/Construct2BR/
Construct 2/3 – Indie Game Brasil	1,380	https://www.facebook.com/groups/construct2indbr/
Desenvolvedores de Games Indie	2,275	https://www.facebook.com/groups/548241321948433/
Desenvolvedores de Jogos Independentes	1,561	https://www.facebook.com/groups/756068824434800/
Games Indie Brasil	1,368	https://www.facebook.com/groups/1135450799818533/
Java Brasil	17,413	https://www.facebook.com/groups/JavaBr/
Python Brasil - Programadores	19,487	https://www.facebook.com/groups/python.brasil/
RPG Maker Brasil	3,646	https://www.facebook.com/groups/418313441606856/
Unity 3D Brasil	16,404	https://www.facebook.com/groups/unity3dbr/
Unity 3D Brasil (1)	7,031	https://www.facebook.com/groups/unity3dbrasil/
Unreal Engine 4 Brasil	5,501	https://www.facebook.com/groups/unrealEngine4Brasil/
Unreal Engine 4 Brasil (Avançado)	1,582	https://www.facebook.com/groups/968146279865089/

Table 2. Online communities of indie game developers (last access: 03/Apr/2017).

Group	Members	Website
Aldeia RPG	16,414	http://www.aldeiarpg.com/
Centro RPG Maker	7,177	http://centrorpg.com/
Game Dev Brasil	4,284	http://www.gamedev.com.br/
Condado Braveheart	1,000	http://www.condadobraveheart.com/

RESULTS & DISCUSSION

The questionnaire remained available for the developer communities for 17 days, during which 35 different questionnaires were answered (without duplicates).

According to the data collected, almost 70% of the independent developers do not use engineering techniques (Fig. 4: Q2). In the circa 30% that do use them, the engineering methods cited are: Design Patterns, MVC (Model-View-Controller), MVVM (Model-View-Viewmodel), MVP (Minimum Viable Product), Scrum, Prototyping, Briefing, and Modeling with Diagrams. (It is not the objective of this

article to discuss the different engineering techniques, but, as they are readily available online, I urge the interested reader to look them up.)

In 2013, Fleury et al. (2014) surveyed the methodologies used for software development in Brazilian game companies, noting that circa 25% did not use any methodology. The absence of software development methodologies was deemed worrying, demonstrating the lack of professionalization of this industry in the country. That survey focused on the actual industry (with a large number of respondents), showing that the problem is not something

endemic of independent developers. In any event, it is a much larger problem in the indie community, even when current literature and previous research strongly advocate the importance of engineering methods.

Among those indie developers that do make use of software engineering, most of them (ca. 74%; Fig. 5) opted for not using agile

methodologies. The so-called “Agile Software Development” are a set of principles for development based on the collaborative effort of self-organizing and cross-functional teams (Wikipedia, 2017a). Among those who do use agile methodologies, Scrum is the most used one (ca. 20%; Fig. 5).

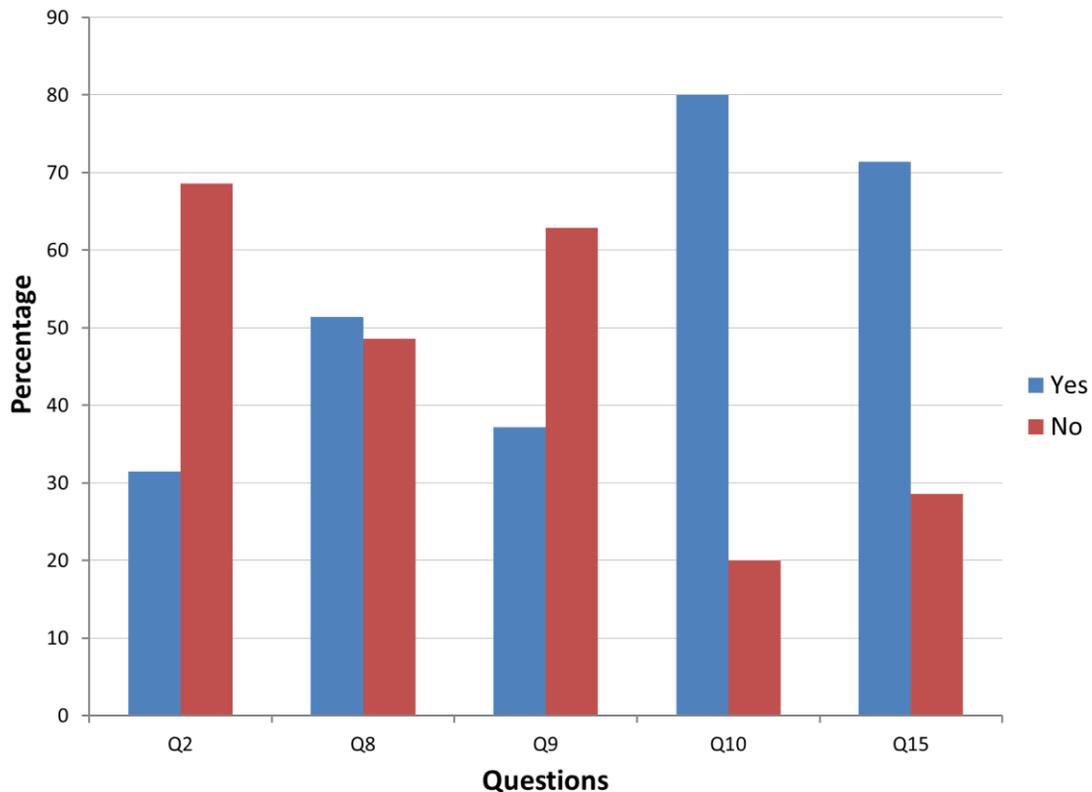


Figure 4. Percentage of answers for Yes/No questions. Question numbers are the same as noted in the Appendix. **Q2:** Do you use engineering methods and techniques in the development process of your game(s)? **Q8:** Do you or your team write a Game Design Document (GDD) at the beginning of your project? **Q9:** Do you and your team follow a system requirements document in game implementation? **Q10:** Do you consider it important to draw up a Game Design Document for your game? **Q15:** Do you consider using software engineering methods important in the process of developing your game(s)?

Another flagrant issue is the lack of a specific professional on the teams who is responsible for the engineering and documentation of the project under development (the Producer mentioned before). By the answers (Fig. 6), having an engineer on

the team is exceedingly rare for independent developers. This position apparently is not deemed important by them, which may explain the rare use of Software Engineering methodologies in their development processes.

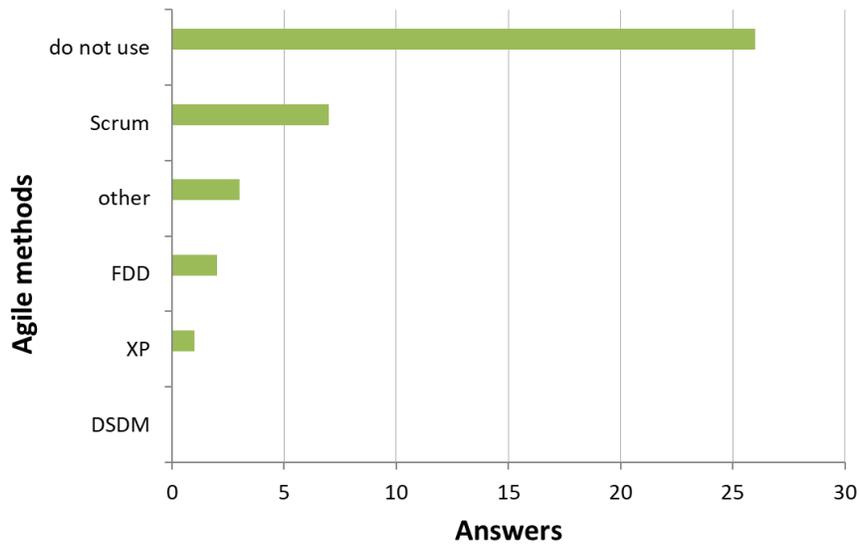


Figure 5. Answers to Q13: Do you use any of these agile methods? Abbreviations: XP = eXtreme Programming; FDD = Feature Driven Development; DSDM = Dynamic System Development Model.

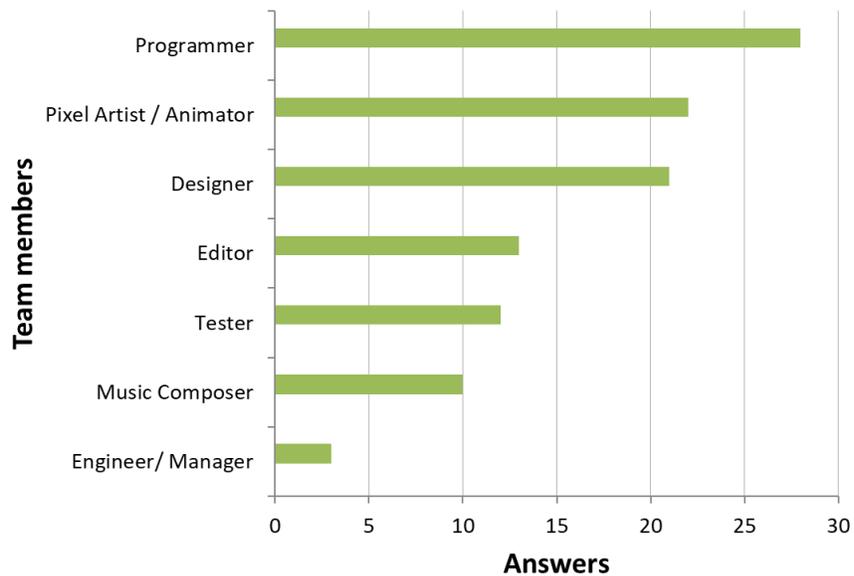


Figure 6. Answers to Q5: Which member(s) make up your development team?

This question can be better understood when we realize that most independent “development teams” actually consist of a single person. About 57% of the interviewed developers work alone, which might explain the usual absence specific software engineering

skills, as most programmers are not specialized in this field. It also explains the anecdotal data on the large number of abandoned projects with exhaustively long production times. Things such as this can be estimated with software metrics, provided there is someone (the

engineer, manager, designer, etc.) with the necessary understanding of Software Engineering (Pressman, 2010).

The elaboration of the GDD, as predicted, was also precarious: it is not made by roughly 50% of the interviewees (Fig. 4: Q8). Failing to elaborate a document with the specificities of the product at the beginning of the project can lead to several problems during the game's development process. It is curious, however, that the importance of the GDD is acknowledged by most developers (80%; Fig. 4: Q10).

Similarly, the importance of using engineering techniques is recognized by most developers (ca. 70%; Fig. 4: Q15), even though only a third of those interviewed (Fig. 4: Q2) actually uses them. This data echoes the above-mentioned survey of Fleury et al. (2014) that evidenced the lack of professionalism by Brazilian game developers.

A common difficulty mentioned (by six respondents) is the dissemination and marketing of the product, which is a key factor, naturally, but one that can be addressed at the beginning of the project based on market risk analysis and estimates of the investments required for a future marketing campaign. That is, this is a problem which can be solved or attenuated by engineering skills. Other difficulties raised were the low investments and scarce incentive to independent development (eight respondents), lack of professionalism (three respondents), and lack of time to finish the game (two respondents).

CONCLUSIONS

From the data obtained here, it can be seen that Brazilian independent game developers still lack professionalism, especially regarding adequate methodologies. Curiously, this is recognized as a problem by the developers themselves. The alarming low usage of Software Engineering techniques also highlights the need for instruction and self-guided research on such methodologies.

Of course, that is not to say that all indie developers in Brazil are in this position or that a lack of professionalism permeates the whole industry in the country. However, the large percentage of developers to which these conclusions apply show that this is a real big issue and the importance of using engineering knowledge to manage and produce quality products cannot be over-emphasized. I hope this serves as a call-to-arms for the indie developers to review their position and start studying and applying concepts from Software Engineering. Citing Velasquez (2009: 30) once again: "*[a] computer game is not just a toy, but a large and complex software project developed by a vast team of professionals*" and should therefore be treated as such.

REFERENCES

- Doolwind.** (2017) Building a strong indie game development team. Available from: <http://www.doolwind.com/blog/building-a-strong-indie-game-development-team/> (Date of access: 22/Jul/2017).
- Fleury, A.; Sakuda, L.O.; Cordeiro, J.H.D.** (2014) 1º Censo da Indústria Brasileira de Jogos Digitais. NPGT/EPUSP, São Paulo.
- Lacerda, E.L. & Selleri, F.** (2012) Um levantamento sobre processos de desenvolvimento de jogos

para redes sociais. Proceedings of SBGames (XI SBGames, Brasília): 77–80.

Lemes, D.O. (2009) Games Independentes: fundamentos metodológicos para criação, planejamento e desenvolvimento de jogos digitais. Pontifícia Universidade Católica de São Paulo, São Paulo. [Unpublished dissertation.]

Morais, F.C. & Silva, C.M. (2009) Desenvolvimento de jogos eletrônicos. e-Xacta 2(2): 11 p.

Pressman, R.S. (2010) Software Engineering: A Practitioner's Approach. Makron Books, São Paulo. [Portuguese (Brazil) edition.]

Sayenko, A. (2015) How (and why) to write a great Game Design Document. Available from: [https://gamedevelopment.tutsplus.com/articles/how](https://gamedevelopment.tutsplus.com/articles/how-and-why-to-write-a-great-game-design-document--cms-23545)

[-and-why-to-write-a-great-game-design-document--cms-23545](https://gamedevelopment.tutsplus.com/articles/how-and-why-to-write-a-great-game-design-document--cms-23545) (Date of access: 22/Jul/2017).

Velasquez, C.E.L. (2009) Modelo de Engenharia de Software para o Desenvolvimento de Jogos e Simulações. Universidade Fernando Pessoa, Porto. [Unpublished dissertation.]

Wikipedia. (2017a) Agile software development. https://en.wikipedia.org/wiki/Agile_software_development (Date of access: 22/Jul/2017).

Wikipedia. (2017b) Indie game. Available from: https://en.wikipedia.org/wiki/Indie_game (Date of access: 22/Jul/2017).

Wikipedia. (2017c) Video game development. Available from: https://en.wikipedia.org/wiki/Video_game_development#Development_team (Date of access: 22/Jul/2017).

APPENDIX: QUESTIONNAIRE

1. Email address: _____

2. Do you use engineering methods and techniques in the development process of your game(s)?

- Yes
 No

3. If you answered “yes” to Question 2, describe which: _____

4. Who is responsible for the game documentation?

Check all that apply.

- Myself
 A specialized person (manager, engineer, etc.)
 Each member documents his/her own work on a single document
 Each member documents his/her own work on separated documents
 No one documents or records activities

5. Which member(s) make up your development team? Check all that apply.

- Programmer
 Pixel Artist / Animation artist
 Designer
 Music Composer
 Tester
 Editor
 Engineer / Manager

6. Describe the platforms, frameworks, game engines you or your team often uses for development: _____

7. Do you work alone or on a team?

- Alone
 Team

8. Do you or your team write a Game Design Document (GDD) at the beginning of your project?

- Yes
 No

9. Do you and your team follow a system requirements document in game implementation?

- Yes
 No

10. Do you consider it important to draw up a Game Design Document for your game?

- Yes
 No

11. If you answered “yes” to Question 10, please explain: _____

12. Describe difficulty(-ies) of the independent game development scene: _____

13. Do you use any of these agile methods? Check all that apply.

- eXtreme Programming (XP)
 Scrum
 Feature Driven Development (FDD)
 Dynamic System Development Model (DSDM)
 Do not use any
 Other

14. In case of other agile methods besides those above, please indicate which: _____

15. Do you consider using software engineering methods important in the process of developing your game(s)?

- Yes
 No

16. Please explain your answer to Question 15:
