Software Engineering Processes in Game Development: a Survey about Brazilian Developers' Experiences

Cristiano Politowski[1]* Daniel de Vargas[1]† Lisandra M. Fontoura[1]‡ Antônio A. Foletto[1]§

[1]Federal University of Santa Maria — Brazil

Abstract

With the increasing participation of digital games in the economy and our society, the attention given to this subject in the academic field has also increased. However, the software engineering field and, more precisely, game development processes seems to be forgotten by researchers. In addition, game developers and big game companies prefer to keep their processes and methodologies to themselves. Studies and professional reports have shown the "ugly face" behind the game industry. Crunch Times and heavy pressure during the development are treated as normal practices in a game developer's life. In this work, we surveyed 58 Brazilian game developers about the relations between game development process and problems in a software engineering context. We sought for answers based on empirical data collected from the questionnaire. The goal was to understand the area and provide insights to improve game development, pointing a direction for future researches. As a result, considering the Brazilian context, this paper presents three main contributions. The first shows that, on projects that used systematic approaches, regardless of the type, result in better products. The second presents that *Delays*, *Unrealistic scope* and *Lack of documentation* are the most common problems faced by game developers. Finally, we describe insights and considerations gathered from developers and literature studies, which may serve as a source of knowledge as well as characterization of the Brazilian game developers.

Keywords: survey, game development process, game developer experience.

1 Introduction

The digital game industry is a billionaire market that has increased its revenue over the years. According to the marketing specialized company Newzoo [31], in 2016 this industry will move about U$99,6 billions, 8,5% more than the last year, with a predict of U$118 billions in 2019.

Although there is not a consensus regarding the nature of digital games (if it is or not a software), game development has particular characteristics and problems which raise its complexity compared to traditional software development [4] [11]. Interviews made by Murphy-Hill *et al.* [29] stated that video game development has significantly differences compared to traditional software development while others authors [3] [11] say that to develop a video game **is** to develop a software. Therefore, due to the higher difficulty of game development, combined with professionals' multidisciplinarity, some authors recommend the use of a Software Engineering (SE) methodology to manage and develop game projects [6].

Despite its decades of existence, development processes in game industry, in general, seems to have not evolved as much as in traditional software community. Through *postmortems*[1] analysis, Petrillo *et al.* [36][37] diagnosed several problems faced by worldwide developers during a game project, being *unrealistic scope*, *feature creep* and *cutting features* the most commons. Moreover, Murphy-Hill *et al.* [29] shows that the game industry, by not using systematic processes, has a lack of maturity. It is a fact that is explicit in the following experience report, gathered by one of the interviewed:

> *"We've got so many specialists on the team, so the kind of planning that you usually do in Agile doesn't work quite so well... You know [specialists] are more concerned about the creative process than an engineering process".*

Still, in the IGDA annual report [46], 52% of the interviewers answered "yes" when asked if *crunch time*[2] was a necessary practice during a game development.

Different from previous works that focused on interviews [17] [5] [29] [32], surveys [30], postmortems analysis [38] [36] [37] [15] and general game industry reports focused on pure qualitative results (more about these works in Section 2) [8] [7] [27] [48] [47] [10] [44] [24], our work offers a new approach, by **surveying Brazilian game developers for relations between the engineering software processes used, problems faced and project's success rate**[3]. To do this, we used three research questions:

1. Is there a relation between the process used and the project's success?
2. Is there a relation between the process used and the problems faced by developers?
3. Is there a relation between the developers' experience and the project's success?

In the absence of a better source, we used empirical data analysis to search for evidences of "why game industry not evolved, in the managerial side, like software industry" and "why so many game developers agree with harmful practices". We believe that, in the context of developing a digital game, if a game is a software, it might have specific elements and characteristics from game development that favor these problems. The Brazilian restricted market was chosen to serve as a preliminary research. The next step is to expand the survey internationally, but first, we decided to test our hypothesis with a small sample.

This paper is structured as follows. Section 2 presents more about the related works. Section 3 explains how we elaborate and

*e-mail: cpolitowski@inf.ufsm.br
†e-mail: ddvargas@inf.ufsm.br
‡e-mail: lisandra@inf.ufsm.br
§e-mail: afoletto@inf.ufsm.br

[1] A document that summarizes the project development experiences, showing the positive and negative outcomes of the development [13].

[2] On game industry, the term *crunch time* refers to periods of extreme work overload, normally in the weeks before deadlines.

[3] "Success", in this case, is related to development time. The concept is described in Section 3.1.

conducted the survey. Section 4 shows the main and secondary results. Section 5 discuss the findings. Section 6 exposes some limitations of this work. Finally, Section 7 concludes with observations regarding the results of the survey and suggest future works.

2 Related Work

Burger-Helmchen *et al.* [5] conducted interviews with eight developers. He divided them in three main communities: users, developers and testers, with the purpose of finding how is the interaction between companies, developers and users' community.

Kasurinen *et al.* [17] interviewed 27 game developers from Finland, ranging four different departments, searching for their expectations regarding design and development tools that they used to use. The results stated that they are satisfied with the tools and practices used, like the use of third party engines that allow the team to focus on game's core functionalities. Moreover, prototyping is the most common practice in the design phase.

Another work made by Kasurinen [16], was a study assessing the video game development from the viewpoint of software engineering where he interviewed 11 companies and conducted a survey to understand the differences between video game and software development. As results, he stated that project management and development tasks are similar, but detailed activities, such as requirements engineering practices, are different. Moreover, he argues that current SE literature does not offer a base to improve the video game development, although Scrum can be suited for this.

O'Hagan *et al.* [32] made a case study with quantitative interviews which found that there is not a good practices model on game development and an approach utilizing ISO/IEC 29110 could be beneficial for the game industry.

Murphy-Hill *et al.* [29] together with a Microsoft SE research group interviewed 14 developers, with at least two year of experience in digital games and traditional software, trying to clarify if game development differs from traditional software development. The results showed that, due to subjective requirements, developing a game is different from traditional software.

Schultz [42] discussed, through bibliographical and documentary research, four topics about video games: traditional cultural industry and digital game industry; market segmentation; business model and video games classification.

Schetinger *et al.* [41] purposed to extend the user stories, an Agile practice, to provide a better documentation and communication among the game development team. They come up with a framework ("three Rs") providing a minimal structure to encapsulate common informations.

Politowski *et al.* [38] analyzed 20 video game postmortems finding that agile (and waterfall in lower degree) process are the most used approach in video game development.

A work that is similar to this one is from Musil *et al.* [30]. He applied an on-line questionnaire to 13 Austrian companies looking for software processes and practices that are used and also questions regarding problems faced by developers. He states that agile software processes, like Scrum, are widely used and the most common problems are *crunch time* and *feature creep*[4].

Our study was influenced by these works mentioned before, but differs greatly by merging the software engineering discipline, more precisely, software processes, together with game development.

There are other empirical studies, developer centered, with data originating from questionnaires. The variables measured are demography, diversity, life quality, job experiences, structures and practices, trends and others metrics more focused on particular aspects of the game industry [8] [7] [27] [48] [47] [10] [44] [24].

3 Method

Our work is based on a Grounded theory [12] [45]. We searched for patterns on empirical data gathered from an on-line survey which was answered by game developers. To make the survey we followed, in a roughly way, the guides provided by Kitchenham [20] [22] [23] [18] [21] [19]. The author described ten steps to conduct a survey since its conception till the results. We used the steps described bellow:

3.1 Setting specific, measurable objectives

Initially, we elaborated a set of four main objectives and three secondary. These items are based on research questions and were used to formulate the question for the questionnaire. The **main objectives** are the following:

1. Gather a list of **processes types** used by developers, regardless the period.
2. Gather the **success rate** of each process type in every project.
3. Gather a list of the most **common problems** faced by game developers in each process type.
4. Gather information about **game developers' experience** in years and if they have ever developed traditional software.

We requested research participants to consider as **"successful"** a project that had few problems, bugs, reworks, was delivered in time (or near) and without a high budget increase. In this case, success has nothing to do with sales, critic or users reception but it is related to development time. Moving forward, the **secondary objectives** are the following:

- Gather game developers' opinions about the importance and adoption of Software Engineering in game development.
- Gather game developers' opinions about the differences in building a game and a traditional software.
- Gather game developers' adoption rate of each type of process.

3.2 Planning and scheduling the survey

Our idea was to gather as many samples as possible in a restricted community. So, we defined the Brazilian game industry as our target and, because of it, the Brazilian game developers. As said before, we decided to work with a small scope because expanding to international community would require more time to apply the questionnaire and also to analyze the data collected. Moreover, to test our hypothesis, a small sample should be sufficient. Although, this limited scope, the data can be reused later by other researcher with the similar interests.

We decided to use an on-line questionnaire, provided by Google Forms, to build the questions, send and receive the developers' answers. The survey was scheduled to range from May 23 to June 6th.

3.3 Designing the survey

We designed the survey in a way it could answer the objectives, as explained above in Section 3.1. With this in mind, we divided the processes in four categories, regarding its nature: Agile, Predictive, Ad-hoc and No-process at all.

A process is **Agile** if the software is built in an iterative approach with continuously process improvement [9]. Developing with Agile is to use small cycles to delivery ready-to-use features each time (iteration) [26]. Examples of this kind of processes are Scrum [43], Extreme Programming (XP) [1], Kanban [33], Adaptative Software Development (ASD) [14] and Feature Driven Development (FDD) [34].

[4] *Feature creep* happens when new functionalities are added during the game development, without proper planning, increasing the project size.

Table 1: Most common problems in game development. Adapted from [37].

Problem	Frequency
Unrealistic scope	75%
Feature Creep	75%
Cutting features	70%
Design problems	65%
Delays	65%
Technological problems	60%
Crunch time	45%
Lack of Documentation	40%
Communication problems	35%
Tool problems	35%
Test problems	35%
Team building	35%
Number of defects	30%
Loss of Professionals	25%
On Over Budget	25%

Predictive processes derive from Waterfall. They are composed by a set of sequential phases and each one of them must be completely finished until the next step. It causes the product value to be completely delivered by the deadline, demanding that requirements have been previously defined [40]. Examples of this kind of processes are Waterfall [2] and Rational Unified Process (RUP) [25].

We defined **Ad-hoc** processes those that are not fitted with Agile or Predictive. Processes that were extremely customized for the company/team needs are considered Ad-hoc too. For those who never used software processes in game projects we defined a last category, called **No-process** or *"code-&-fix approach"*[5].

The whole questionnaire is divided in *five sessions*. To start, in **session #1**, we asked the developer his/her *academic and technical background*, opinions about the *importance of software engineering* and differences between *developing a software and a digital game*. The remaining four sessions (**sessions #2, #3, #4, #5**) are for each process category: Agile, Predictive, Ad-Hoc and No-Process.

First we asked if the developer had experience developing games with the particular process type, if the answer was "yes", then the respondent was redirected to a new session where there were specific questions regarding that process, like *process types used*, *number of projects*, *success rate* and *problems faced*. If the respondent does not have a determined process experience, he/she will be redirected to the next category. This flow can be better visualized in Figure 1 and in the complete questionnaire[6].

In order to populate the questionnaire with problems related to game development, we used the list provided by Petrillo *et al.* [37] [36], described in Table 1, in which he gathered, from twenty post-mortem analysis, the most common problems reported by game developers. In each questionnaire session, the respondent should mark the three most common problems that occurred during all his/her experience as game developer. The problems are listed in Table 3.

3.4 Validating the instrument

The form validation occurred in two stages. First, we asked software engineering professors to analyze the questions' correctness. Second, we sent the questionnaire to game developers to analyze the usability and understanding.

Table 2: Relation between process used and the project success

Agile		Predictive		Ad-hoc		No-process	
Failure	Sucess	Failure	Sucess	Failure	Sucess	Failure	Sucess
56	236	22	86	7	54	39	76
19,18%	80,82%	20,37%	79,63%	11,48%	88,52%	33,91%	66,09%

3.5 Selecting participants

We started this step by searching for game developers and software **associations** using the Google search tool. We searched for the following strings: *"Associações de desenvolvedores de jogos de <estado>"*, *"APL audiovisual do <estado/região>"*, *"APL software do <estado/região>"* and *"Festival de desenvolvimento de jogos do Brasil"*. As result, we tabulated 125 different associations, with name, page url, summary and observations on each one. After, using this list, we sought for game developers companies. The result was 347 different companies[7], tabulated with "name", "page", "url", "state", "city", "contact", "description" and "observations". The overall companies, grouped by states, can be visualized in Figure 2.

The next step was to verify every company and check if it was active or inactive. This verification was made through a contact with the company by email or social network. A total of 253 firms answered, of which 236 said that they were active and 17 inactive. The remaining 94 companies did not reply.

3.6 Administering and scoring the instrument

Afterwards, we sent emails containing the questionnaire to all the active companies. Unfortunately, for technical reasons, 36 emails did not reach the companies. Moreover, we sent the questionnaire to developer groups in social networks, like "Game Developers" on *Linkedin* and "Indie Game Developers Brasil" e "Game Experience Brazil" on *Facebook*.

4 Results

Although the high number of Brazilian game companies, we got only 62 developers replies. From those replies, one sample was noisy (different values from the expected) and in three samples the answers contradicted each other. Even though the final sample size was 58, we obtained very interesting insights that are described below.

In Figure 3 we can see the success rate in every process type. Projects using Ad-hoc processes, with a success rate of 88,53%, represent the best result, followed by Agile with 80,82%, Predictive with 79,63% and lastly No-process, with 66,09%.

When looking for the number of projects in each process category we noticed a disparity. There are more projects related being Agile than the other types, as seen in Table 2.

Other correlation is related to the process used and the problems faced by developers. In Table 3 is listed the 15 problems and its occurrence in each process type. Moreover, the last column shows the total occurrences of each problem in all the projects analyzed. In **Agile**, the most common problems are *Unrealistic scope* with 15,76%, *Delays* with 14,55% and *Communication problems* with 10,91%. **Predictive** processes present *Delays* with 20,00%; *Unrealistic scope*, *Lack of Documentation*, *Communication problems*, *Test problems* all together with 8,57%. In **Ad-hoc** processes the most common problems are *Delays* with 17,14%, *Lack of Documentation* with 14,29% and *Unrealistic scope* and *Cutting features* together with 11,43%. Lastly, when **No-process** is used, the three

[5] Actually, it is an *anti-pattern*. It is called "Cowboy coding" too, where the developer have autonomy over the development process [49].

[6] The complete questionnaire can be visualized on the project's website: *http://polako.github.io/gamedev-process-survey/survey-form-export.pdf*

[7] The list of video game associations and companies, in CSV format, can be visualized on the project's website: *http://polako.github.io/gamedev-process-survey*

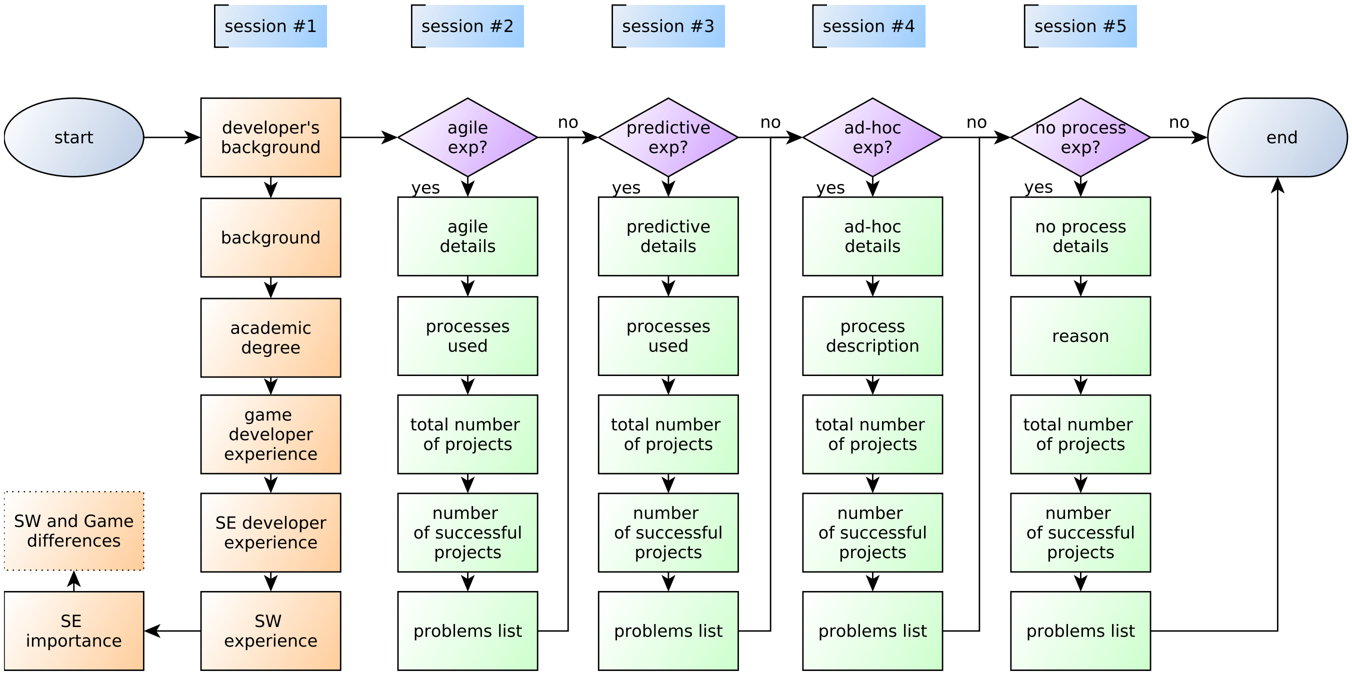

Figure 1: Questionnaire design flow. Dotted lines represent optional questions.

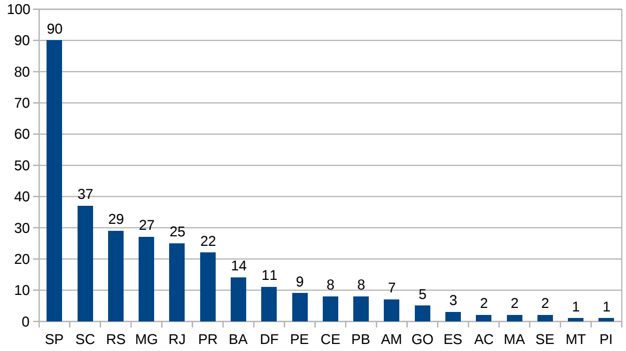

Figure 2: List of Brazilian game companies by state.

major problems are *Unrealistic scope* with 14,94%, *Lack of Documentation* with 11,49% and *Delays* and *Number of defects* with 9,20%.

In addition, if we consider all the projects, regardless the process nature, we have *Delays* with 60,88% followed by *Unrealistic scope* with 50,70% and *Lack of Documentation* with 44,05%. All the problems, grouped by process type, are shown in Figure 4.

Two questions were made regarding the developers experience: one about the experience time developing games and another about experience with traditional software development. The Figure 5 shows the relation between the years developing games and the success rate in projects.

The Figure 6 shows if the experience working with traditional applications can influence the success rate in game projects. Surprisingly, developers without experience developing traditional software reported a greater success using No-process approach than others that have experience. Nonetheless, the difference is not big enough to make assumptions.

Despite the main results, other interesting informations could be

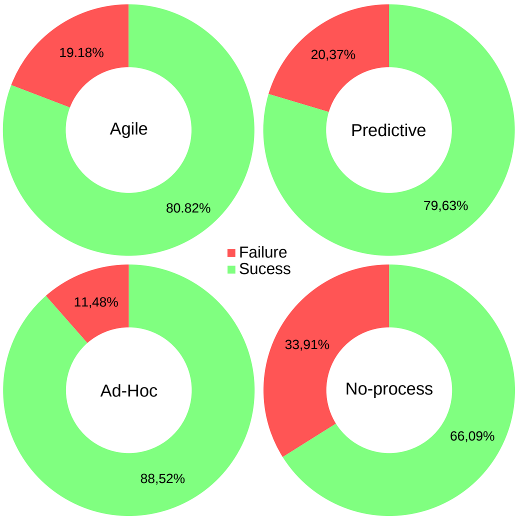

Figure 3: Success and Failure in Brazilian game development according to process type.

Table 3: Relation between process used and the problems faced by developers.

Problem	Agile	Predictive	Ad-hoc	No-process	Frequency
Delays	14,55%	20,00%	17,14%	9,20%	60,88%
Unrealistic scope	15,76%	8,57%	11,43%	14,94%	50,70%
Lack of Doc.	9,70%	8,57%	14,29%	11,49%	44,05%
Cutting features	7,27%	5,71%	11,43%	5,75%	30,16%
Design problems	6,06%	7,14%	8,57%	6,90%	28,67%
Com. problems	10,91%	8,57%	2,86%	5,75%	28,08%
Crunch time	6,67%	7,14%	5,71%	8,05%	27,57%
Feature Creep	4,85%	5,71%	8,57%	6,90%	26,03%
Test problems	7,27%	8,57%	2,86%	6,90%	25,60%
Num. of defects	1,21%	1,43%	8,57%	9,20%	20,41%
Over Budget	4,85%	2,86%	2,86%	6,90%	17,46%
Team building	1,82%	5,71%	2,86%	3,45%	13,84%
Loss of Prof.	5,45%	5,71%	0,00%	1,15%	12,32%
Tech. problems	2,42%	1,43%	0,00%	3,45%	7,30%
Tool problems	1,21%	2,86%	2,86%	0,00%	6,93%

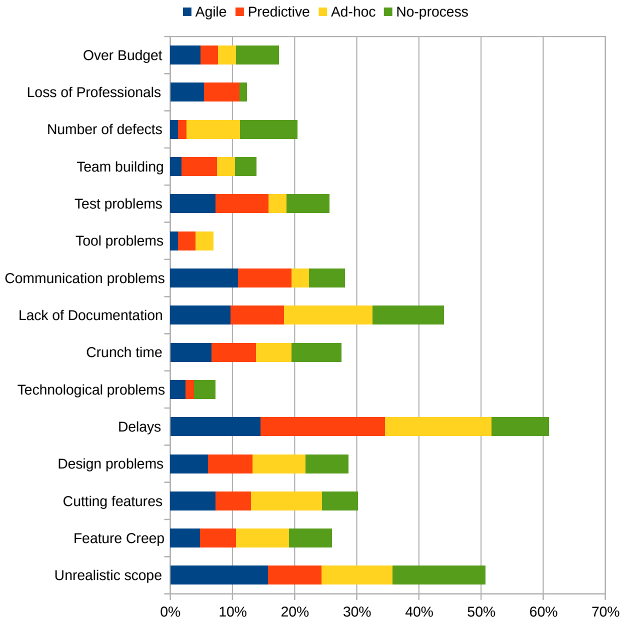

Figure 4: Aggregated data showing the most frequently problems grouped by process type.

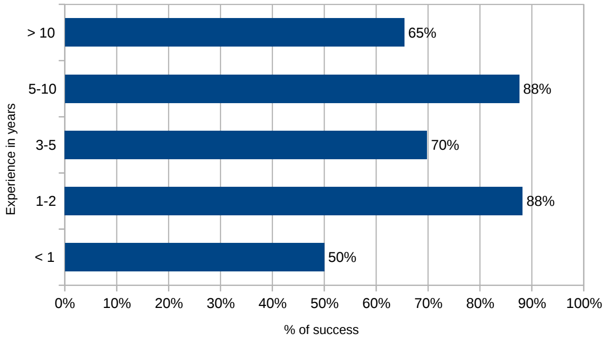

Figure 5: Relation between the developers' experience developing games and the percentage of projects success.

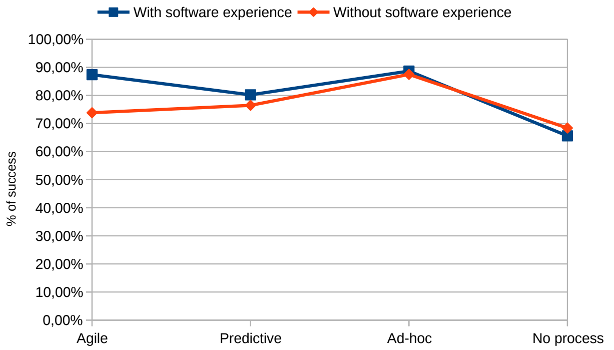

Figure 6: Relation between the developers' experience regarding traditional software and the percentage of projects success.

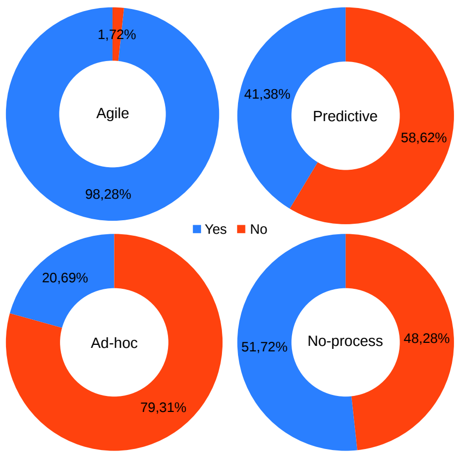

Figure 7: Game developers' experience using Agile, Predictive, Ad-Hoc and No-process.

highlighted. The first one states that the most common approach for developing games is with Agile, with 98,28% (57 samples) of respondents reporting at least one project made using this process. With more than a half of answers, No-process is the second most used method, with 51,72% (30 samples). Predictive with 41,38% (24 samples) comes right after and lastly, Ad-hoc, with 20,69% (12 samples). The data is in Figure 7.

Regarding SE, two questions were made: on the first one, it was asked about the importance of this field (SE) for game development; on the second one, it was asked how frequently SE practices are applied during a game project. The alternatives were in 5-points scale, as Figure 8 state.

5 Discussion

The game development seems to be best suitable with a customized approach, different from traditional software. This is evidenced by the highest success rate showed by Ad-hoc processes type. Yet, pure traditional software methods like Agile and Predictive (Waterfall) had also a high success rate, confirming the results gathered by Politowski *et al.* [38]. A clarified result is regarding the success rate of No-process approach, being the lowest one with around tree times less effectiveness compared to Ad-hoc. Although we expected a clearer difference among the processes types considering the success rate, it reveals that, even not being a standard in video game development nor a well established practice, a systematic approach appears to deliver better products.

Analyzing the most common problems reported by developers

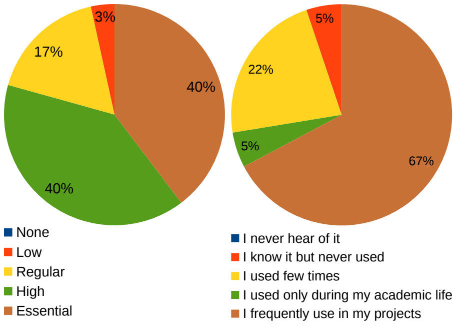

Figure 8: (left) How much is the importance of SE in game development. (right) How frequently SE is used.

Table 4: Correlations between problems in each process type.

Correlation	Agile	Predictive	Ad-hoc	No-process
Agile	100,00%	**79,13%**	60,84%	67,95%
Predictive	**79,13%**	100,00%	62,84%	39,83%
Ad-hoc	60,84%	62,84%	100,00%	**71,35%**
No-process	67,95%	39,83%	**71,35%**	100,00%

together with those reported by Petrillo [37], we may state that issues regarding video game *scope* are the major source of headaches for game developers. Normally, this problem comes together with *feature creep*, however, it is not the case because feature creep appears only in the ninth place with 26,03%. This is going in the opposite direction of what happens in Austrian game industry, as stated by Musil [30], with *crunch time* and *feature creep* as being the most common problems.

The second place problem, *Lack of documentation*, has high occurrence even in predictive methods (8,57%). Yet, it is in ad-hoc (14,29%) and no-process (11,49%) approaches where the numbers are greater. It shows that besides the high adoption of agile approaches in software game development, documentation is an important artifact and should be considered as a required step in video games project management.

Even though, *Delays* are the most common problem related by the respondents, with 60,88% of frequency. They can be caused by several factors. The root cause may be related with the requirements phase. This step is not trivial in a video game project, notwithstanding the only real requirement is that the game must be “fun” [39] [35] [32] [28]. Better prototyping or brainstorming phase together with a special attention to documentation along the project life cycle may mitigate this problem.

By grouping the most common problems in each process type, we can make a correlation between them (Table 4). Surprisingly, Agile and Predictive share similar problems, with a correlation of 79,13%. In the other hand, problems are more alike in Ad-hoc and No-process, with correlation of 71,35%.

Concerning developers experience and projects success rate, there is not a clear relation between this two variables. The success rate informed by developers with less than one year (50%) of experience is similar to developers with the highest experience (65,41%). Still, developers with no experience in traditional software show better results (success) utilizing No-process approach.

Strengthening the results provided by Politowski *et al.* [38] and Musil [30], agile appears as (by far) the most used process in video game development. Since its beginnings in mid-2001, the agile culture has been spreading fast and, at a slower speed, game developers are adopting those concepts. The unpredictability and multidisciplinarity of video game development scenario appears to fit better in small cycles of continuing delivery.

Another surprisingly result was the developers’ concern regarding software engineering field. The majority of respondents (80%) considered the discipline very important while 67% frequently use SE practices. We expected a high number of developers unfamiliar with this area, but considering that more than half of samples has a Computer Science background, the relation becomes clearer.

Also, some open questions were made in the questionnaire. One of them is related to game developers’ opinions about differences between building a traditional software and a video game. The large majority of respondents said that there are differences among them and only 5 developers said otherwise. Among these answers, interesting viewpoints can be highlighted:

- Traditional software has a linear development while games are more dynamic;
- A traditional software is thought to be eternal while video games have short life;
- Unlike traditional software that uses a “definition of done”, a game “working” is just the beginning of the job;
- Multidisciplinarity is stated as the most different aspect between software and games;
- The game creation process involves more user testing than software;
- Game engines restrict the use of some patterns in favor of better productivity;
- There is a higher coupling in game development pipeline compared to traditional software development;
- In software, it is easier to translate a requirement list to tasks while in game development the search for the fun factor involves several features combinations (macro-feature).

With respect to ad-hoc processes, this questionnaire section asked the developer how was this kind of process, the determined steps and practices. The next list shows the more relevant answers (each item is a summary of an ad-hoc process described):

- *“Full autonomy during development with experienced team”*;
- *“Initial definition with artistic freedom plus constant changes”*;
- *“Product objective definition, brainstorming, project, prototyping, validation, project, prototyping, validation, tests and bugs corrections, postmortem and maintenance”*;
- *“Design project and features definition”*;
- *“Four production lines: creation, assets, assemble and test. Each one has a defined process”*;
- *“Fixed quality: zero bugs. Minimal scope stipulated (MVP) but, after this point, flexible and validated by final users. Deadline with 25% of flexibility”*;
- *“Milestones delivered to users without defined scope”*.

Lastly, it was asked why developers do not use a systematic approach to develop their video games. The majority of answers were due to the lack of *knowledge* or *experience*, followed by *short time* and *small teams*. Although, a portion of developers appears to consider SE relevant for game development, these results evidence that a good amount of projects being developed with no systematic approach.

6 Treats of Validity

There are some limitations in this work. First, the sample analyzed is small and, for this reason, hard to make a generalization. Nevertheless, if we think in a small context, like “Brazilian game developers”, the results presented here look more reliable and similar with reality. Second, the respondents are from different video game groups, genre expertise, team size, project size, among many others. It was defined as a criteria that the developer must have participated in at least one game project. Due this, the target group may seem a bit large. Lastly, although the data passed by a noise removal step, the answers may contain bias, compromising the statistics.

7 Conclusions

This work presented a survey about video game developers experiences regarding software engineering processes. We sought for patterns and correlations in empirical data, gathered from an online questionnaire sent to Brazilian video game developers.

In this paper we presented three primary contributions gathered from developers descriptions of their previous experiences developing video games. The data shows that, in a Brazilian context, **projects that used a systematic approach, regardless of the type, resulted in better products**.

Although not as accurate as literature argues, ***Delays*, *Unrealistic scope* and *Lack of documentation* are the most common problems faced by Brazilian game developers**. Moreover, a correlation greater than 70% was noted between problems with Agile and Predictive and with Ad-Hoc and No-process.

Considering the lack of specialized literature, the results presented here can be a source of knowledge about video game development and SE process adoption.

The next steps of the research is to extend this work by expanding the scope, define a new variables set, make use of interviews and other kinds of empirical methods to extract more about video game development processes.

Acknowledgment

We would like to thank all companies and respondents for their participation. Also, thank the CAPES and CNPq for the financial support. This project was developed in the context of the Brazilian Army Strategic Project ASTROS 2020.

References

[1] K. Beck. *Extreme Programming Explained: Embrace Change*. An Alan R. Apt Book Series. Addison-Wesley, 2000.

[2] T. E. Bell and T. A. Thayer. Software requirements: Are they really a problem? In *Proceedings of the 2Nd International Conference on Software Engineering*, ICSE ’76, pages 61–68, Los Alamitos, CA, USA, 1976. IEEE Computer Society Press.

[3] E. Bethke. *Game development and production*. Wordware Publishing, Inc., 2003.

[4] J. Blow. Game development: Harder than you think. *Queue*, 1(10):28, 2004.

[5] T. Burger-Helmchen and P. Cohendet. User communities and social software in the video game industry. *Long Range Planning*, 44(5):317–343, 2011.

[6] D. Callele, E. Neufeld, and K. Schneider. Requirements engineering and the creative process in the video game industry. In *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*, pages 240–250. IEEE, 2005.

[7] K. Edwards, J. Weststar, W. Meloni, C. Pearce, and M.-J. Legault. Developer satisfaction survey 2014: Summary report. Technical report, 2014.

[8] A. C. C. Fleury and D. N. Nakano. Mapeamento da Indústria Brasileira e Global de Jogos Digitais. Technical report, 2014.

[9] M. Fowler. The new methodology. *Wuhan University Journal of Natural Sciences*, 6(1-2):12–24, 2001.

[10] Game Developers Conference. GDC 2016 State of the Industry Report. Technical report, 2016.

[11] A. Gershenfeld, M. Loparco, and C. Barajas. *Game plan: the insider’s guide to breaking in and succeeding in the computer and video game business*. St. Martin’ s Griffin Press, New York, 2003.

[12] B. G. Glaser and A. L. Strauss. The discovery of grouded theory. *Chicago (US): Aldine*, 1967.

[13] W. Hamann. Goodbye postmortems, hello critical stage analysis. *Gamasutra - The Art & Business of Making Games*, July 2003.

[14] J. Highsmith. *Adaptive Software Development: A Collaborative Approach to Managing Complex Systems*. Dorset House eBooks. Pearson Education, 2013.

[15] M. W. Jr, P. Sathiyanarayanan, M. Nagappan, T. Zimmermann, and C. Bird. ”what went right and what went wrong”: An analysis of 155 postmortems from game development. In *Proceedings of the 38th International Conference on Software Engineering (ICSE 2016 SEIP Track)*. ACM – Association for Computing Machinery, May 2016.

[16] J. Kasurinen. Games as Software – Similarities and Differences between the Implementation Projects. In *International Conference on Computer Systems and Technologies - CompSysTech’16 Games*, 2016.

[17] J. Kasurinen, J.-P. Strandén, and K. Smolander. What do game developers expect from development and design tools? In *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, pages 36–41. ACM, 2013.

[18] B. Kitchenham. Principles of Survey Research Part 4: Questionnaire Evaluation. 27(3):20–23, 2002.

[19] B. Kitchenham and S. L. Pfleeger. Principles of Survey Research Part 6: Data Analysis. pages 24–27.

[20] B. Kitchenham and S. L. Pfleeger. Principles of Survey Research Part 1: Turning Lemons into Lemonade. *ACM SIGSOFT Software Engineering Notes*, 26(6):16, 2001.

[21] B. Kitchenham, S. L. Pfleeger, and A. S. Development. Principles of Survey Research Part 5: Populations and Samples. 27(5):17–20, 2002.

[22] B. A. Kitchenham and S. L. Pfleeger. Principles of Survey Research Part 2 : Designing a Survey. 27(1):18–20, 2002.

[23] B. A. Kitchenham and S. L. Pfleeger. Principles of Survey Research Part 3: Constructing a Survey Instrument. 27(2):20–24, 2002.

[24] Krimm240. /r/ GameDev 2016 Survey Results!: gamedev, 2016.

[25] P. Kruchten. *The Rational Unified Process: An Introduction*. The Addison-Wesley object technology series. Addison-Wesley, 2004.

[26] C. Larman and V. R. Basili. Iterative and incremental development: A brief history. *Computer*, (6):47–56, 2003.

[27] M.-j. Legault and J. Weststar. WORKING TIME AMONG VIDEO GAME DEVELOPERS, 2004-14. Technical report, 2015.

[28] C. Lewis and J. Whitehead. The whats and the whys of games and software engineering. In *Proceedings of the 1st International Workshop on Games and Software Engineering*, pages 1–4. ACM, 2011.

[29] E. Murphy-Hill, T. Zimmermann, and N. Nagappan. Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? In *Proceedings of the 36th International Conference on Software Engineering (ICSE 2014)*. ACM, June 2014.

[30] J. Musil, A. Schweda, D. Winkler, and S. Biffl. A survey on the state of the practice in video game software development. Technical report, Technical report, QSE-IFS-10/04, TU Wien, 2010.

[31] NEWZOO. The global games market reaches $99.6 billion in 2016, mobile generating 37%. https://goo.gl/fzFb7F, Apr. 2016.

[32] A. O. O’Hagan and R. V. O’Connor. *Systems, Software and Services Process Improvement: 22nd European Conference, EuroSPI 2015, Ankara, Turkey, September 30 – October 2, 2015. Proceedings*, chapter Towards an Understanding of Game Software Development Processes: A Case Study, pages 3–16. Springer International Publishing, Cham, 2015.

[33] T. Ohno. *Toyota Production System: Beyond Large-Scale Production*. Taylor & Francis, 1988.

[34] S. Palmer and J. Felsing. *A Practical Guide to Feature-driven Development*. The Coad series. Prentice Hall PTR, 2002.

[35] F. Petrillo and M. Pimenta. Is agility out there?: agile practices in game development. In *Proceedings of the 28th ACM International*

Conference on Design of Communication, pages 9–15. ACM, 2010.

[36] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich. Houston, we have a problem...: a survey of actual problems in computer games development. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 707–711. ACM, 2008.

[37] F. Petrillo, M. Pimenta, F. Trindade, and C. Dietrich. What went wrong? a survey of problems in game development. *Computers in Entertainment (CIE)*, 7(1):13, 2009.

[38] C. Politowski, L. Fontoura, F. Petrillo, and Y.-G. Guéhéneuc. Are the old days gone?: A survey on actual software engineering processes in video game industry. In *Proceedings of the 5th International Workshop on Games and Software Engineering*, GAS '16, pages 22–28, New York, NY, USA, 2016. ACM.

[39] R. Ramadan and Y. Widyani. Game development life cycle guidelines. In *Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on*, pages 95–100. IEEE, 2013.

[40] W. W. Royce. Managing the development of large software systems. In *proceedings of IEEE WESCON*, volume 26, pages 328–388. Los Angeles, 1970.

[41] V. Schetinger, C. Souza, L. M. Fontoura, and C. T. Pozzer. User stories as actives for game development. In *SBGames 2011*, pages 1–4, 2011.

[42] C. Schultz. Fragmentação no Mercado de Jogos Digitais. In *SBGames 2014*, pages 1111–1118, 2014.

[43] K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. Series in agile software development. Pearson Education International, 2002.

[44] SIOUX, BLEND, and ESPM. Pesquisa Game Brasil 2016. Technical report, 2016.

[45] A. Strauss, J. Corbin, et al. *Basics of qualitative research*, volume 15. Newbury Park, CA: Sage, 1990.

[46] J. Weststar and M. Andrei-Gedja. Developer Satisfaction Survey 2015 Industry Trends and Future Outlook Report. Technical report, 2016.

[47] J. Weststar and M. Andrei-Gedja. Developer Satisfaction Survey 2015 Industry Trends and Future Outlook Report. Technical report, 2016.

[48] J. Weststar and M.-J. Legault. Developer satisfaction survey 2015: Summary report. Technical report, 2015.

[49] Wikipedia. Cowboy coding — wikipedia, the free encyclopedia, 2016. [Online; accessed 11-July-2016].