Are the Old Days Gone? A Survey on Actual Software Engineering Processes in Video Game Industry

Cristiano Politowski, Lisandra Fontoura
Federal University of Santa Maria
Santa Maria, Brazil
{cpolitowski,lisandra}@inf.ufsm.br

Fabio Petrillo, Yann-Gaël Guéhéneuc
École Polytechnique de Montréal
Montréal, Canada
fabio@petrillo.com,
yann-gael.gueheneuc@polymtl.ca

ABSTRACT

In the past 10 years, several researches studied video game development process who proposed approaches to improve the way how games are developed. These approaches usually adopt agile methodologies because of claims that traditional practices and the waterfall process are gone. However, are the "old days" really gone in the game industry?

In this paper, we present a survey of software engineering processes in video game industry from postmortem project analyses. We analyzed 20 postmortems from Gamasutra Portal. We extracted their processes and modelled them through using the Business Process Model and Notation (BPMN).

This work presents three main contributions. First, a postmortem analysis methodology to identify and extract project processes. Second, the study main result: **the "old days" are gone, but not completely**. **Iterative practices** are increasing and are applied to at least **65% of projects** in which **45% of this projects** explicitly adopted Agile practices. However, **waterfall** process is still applied at least **30% of projects**. Finally, we discuss some implications, directions and opportunities for video game development community.

Keywords

Software engineering process; Game development, Postmortem; Survey

1. INTRODUCTION

Game development is an extremely complex activity [10]. A study on game industry problems [29] concluded that game projects does not suffer from technological problems but essentially from management and process. To alleviate theses issues, several academic and partitioner studies have been conducted for years, especially on the adoption of agile game development [15, 16, 28]. Moreover, a recent systematic literature review [27] observed that there are substantially fewer industrial studies about game development processes and claims that agile processes are appropriate when innovation and speed to market are vital in game development. In the same direction, a developer gave us some interesting observations in a recent postmortem about game development [9]:

> *"**The old days are gone**. You can't expect producers or leads to come up with a huge waterfall of everything they thought would get done over the next three years. In the game development business, it's insane to think you have any insight into what your team will be doing one year from now. You can set major milestones with hard dates, but filling in all the details between those points is an exercise in futility."*

Nevertheless, is this claim general or the developer's bias vision? **Are "the old days" really gone in video game industry?** For the purpose of answering this question and studying game development processes in industrial context, we conduct a survey of the software engineering processes in video game industry from postmortem project analyses, modelling them using Business Process Model and Notation (BPMN) [26].

This paper is structured as follows. Section 2 summarizes briefly the main concepts and definitions about game development processes and practices used in this work. Section 3 presents our research methodology to analyse and collect data from project postmortems. Section 4 shows the results of study, providing project process diagrams, quantitative and qualitative results, and discussing our findings. Section 5 presents the related work. Section 4.3 shows threats of validity. Finally, Section 6 summarizes this survey and discusses some implications and future work opportunities.

2. BACKGROUND

In this section, we define briefly some concepts used in this work to provide a single interpretation. First, we define concepts about game development processes described in the literature. Then we describe postmortems in game development.

2.1 Game Development Process

Game development processes can be classified into four main categories: waterfall, iterative, hybrid, and ad-hoc [1, 27]. In this work, we define each category as follows:

Waterfall [30] (or predictive) is a sequential process in which a next phase is started only if the previous phase is

GAS'16, May 16 2016, Austin, TX, USA

ISBN 978-1-4503-4160-8/16/05. . . $15.00
DOI: http://dx.doi.org/10.1145/2896958.2896960

completely finished, delivering business value all at once. This is the traditional game development process, requiring explicit requirement assessments followed by orderly and precise problem solving procedures.

Iterative [20] is a process which consists to develop a software by repeating short-cycles to deliver a ready-to-use feature each time. Agile software methodology follows this iterative approach, improving continuously and systematically its processes and practices [8].

Hybrid is a combination of waterfall and iterative processes in the same project. Typically, the waterfall strategy is used during pre/post-production and the iterative is applied during the production phase.

Ad-Hoc is a process that is created only for a specific project, without a previous definition. In ad-hoc process, activities are defined on demand and the process changes to respond to punctual and contextual issues.

2.2 Game Development Postmortems

The term *postmortem* designates a document that summarizes the project development experiences, with a strong emphasis on the positive and negative outcomes of the development [11]. It is commonly done right after the project finishes, by managers or senior project participants [5]. It is a important tools for knowledge management [3], from which the group can learn from its own experiences and plan future projects. Postmortem analysis can be so revealing that some authors [3] argue that no project should finish without postmortem.

Postmortems are much used in the game industry. Many game Web sites devote entire sections to present these documents, such as *Gamasutra* (http://www.gamasutra.com) and *Gamedev* (http://www.gamedev.net). These postmortems pertain to a variety of development teams profiles and projects, varying from few developers in small projects to dozens of developers in five-year-long projects.

The postmortems published by *Gamasutra* mainly follow the structure proposed by the *Open Letter Template* [24], which is composed of three sections. The first section summarizes the project and presents some important outcomes of the development. The next two sections discuss the most interesting of game development:

- **What went right:** it discusses the *best practices* adopted by developers, solutions, improvements, and project management decisions that have improved the efficiency of the team. All these aspects are critical elements to be used in planning future projects.
- **What went wrong:** it discusses difficulties, pitfalls, and mistakes experienced by the development team in the project, both technical and managerial.

The information contained in postmortems constitute a piece of knowledge that can be reused by any development team, including examples and real life development experiences. They allow knowledge sharing and are useful for planning future projects or process improvements.

3. RESEARCH METHODOLOGY

To conduct our survey and answer our research question, we elaborated a research method to analyze postmortems and extract the processes used in game development. Our method contains several steps whose output is the input for a next step. In this section, we present these steps, which are modeled in Figure 1.

Table 1: List of analyzed postmortems and notes.

#	Postmortems	# notes
1	Brutal Legend	16
2	Kingdoms of Amalur: Reckoning	19
3	Caseys Contraptions	18
4	Sins of a Solar Empire	27
5	Amnesia: A Machine for Pigs	27
6	City Conquest	26
7	Baldurs Gate Enhanced Edition	26
8	Trine	49
9	Natural Selection 2	28
10	The Path	35
11	Dust An Elysian Tail	28
12	Anomaly Warzone Earth	42
13	Aaaaa! – A Reckless Disregard for Gravity	14
14	Scooby-Doo First Frights	25
15	Spider-Man	32
16	Deadliest Warrior	12
17	Zack Zero	23
18	God of War Ascension	22
19	Electronic Symphony The Untold Story	8
20	Guacamelee	21
	Total	498

Step 1. Getting postmortems: first, we searched in Gamasutra by all the postmortems ranged from 2010 until now. Our focus was gather the latest projects, so postmortems prior to 2010 were discarded. We collected a total of 63 articles[1].

Step 2. Keyword definition: as done by Petrillo *et al.* [29], we limited our study in 20 postmortems, defining keywords for filtering, searching by concepts on software engineering [2, 7, 8, 12, 17, 21, 22, 33] and glossaries [4, 13, 19, 31, 32, 34, 35]. After that, we got 682 unique terms.

Step 3. Postmortems filtering: next, utilizing the text mining software DocFetcher[2] on the terms got from step 2, we selected the 20 postmortems (Table 1) with greater number of found terms.

Step 4. Postmortem analysis: with a collection of selected postmortems, we read every postmortem thoroughly, searching for process elements like *activities*, *roles*, *artifacts*, *practices* and all other elements related to software engineering process and management. As a result, we extracted a total of 498 notes and respective quotations from the 20 postmortem analyses.

Step 5. Game process modeling: we analyzed the postmortem notes to build a process model for each project, modeling activities or process details, using the Business Process Model Notation (BPMN) version 2.0 [26]. To guide this modeling process, we defined a meta-model in Figure 2. Following this steps, we created a process model for each postmortem in which we can observe clearly its characteristics and highlights the work-flow for each project. The meta-model consist in several elements, described on Table 2. Furthermore, we highlighted in red some **problematic elements** explicitly declared by a postmortem author.

Step 6. Postmortem Process Analysis: using the

[1] The collection of our raw data and diagrams are available in http://gas2016.github.io

[2] http://docfetcher.sourceforge.net/en/index.html

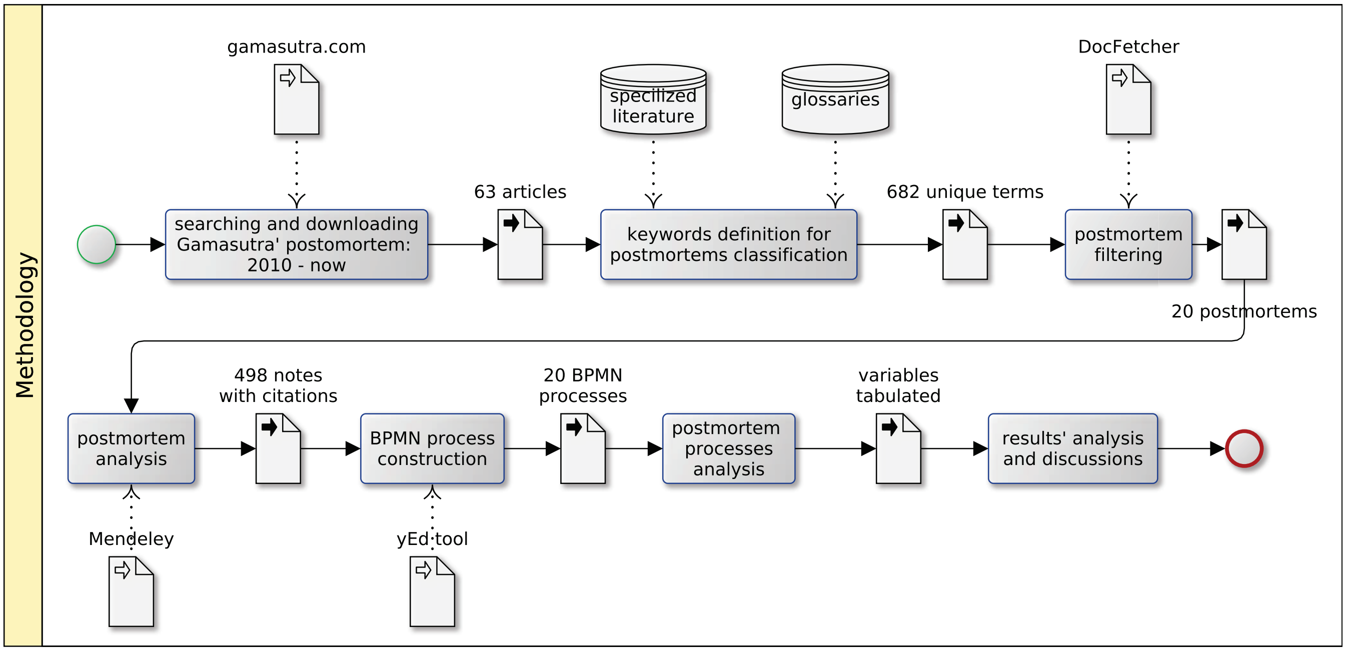

Figure 1: The research method in BPMN notation.

process diagrams and quotations, we analyzed each project, establishing two essential variables to answer our research question. The first variable is **process**, which shows in what process category (Section 2.1) a process is classified. The second is **agile**, a boolean variable that shows if a project follows agile practices. Finally, we collected process data and organized them in Table 3.

Following these steps systematically, we produced several processes used in the video game industry. These results and their discussions are presented in the next section.

4. RESULTS

We analyze the postmortem diagrams and quotations, categorizing each project as *waterfall, iterative, hybrid, or ad-hoc* based on the definitions presented in Section 2. We organize this information in Table 3. Evaluating this data, we found that 55% of projects (11/20) adopt an iterative process; 30% of projects (6/20) use waterfall process; 10% of projects (2/20) are hybrid; 5% (1/20) apply ad-hoc practices. These results are presented in Figure 3.

4.1 Typical Process Models

To present each process category, we selected four typical project models: a hybrid (Brutal Legend - Figure 4), an iterative (Kingdoms of Amalur: Reckoning - Figure 5) , a ad-hoc (Aaaaa! A Reckless Disregard for Gravity - Figure 6), and finally a waterfall (Scooby-Doo First Frights - Figure 7)[3]. The next paragraphs highlight some common characteristics typically found in each process category.

Composed by a combination of iterative and waterfall processes, Figure 4 is an example of hybrid process. This process has a remarkable characteristic: the process is clearly separated in pre-production, production, and post-production.

Table 2: Description of meta-model elements.

Element	Description	Examples
Team characteristics	Team characteristics concern team peculiarities. Any detail, highlighted by the author, about the team in development process.	*multi-disciplinary teams*, *novice team* and *full-stack developer*.
Sub-process	Sub-process highlights a division or a special feature in a workflow. For example, an iterative sub-process represents that the flow was repeated many times. When the word "production" is written on a element label, it means a waterfall-like method.	*pre-production*, *production (iterative)*, or others ad-hoc workflows.
Repositories	Repositories are used whenever a source of information or a set of ideas was stored.	*idea pool* and *bug repository*.
Artifacts	Artifacts are pieces of work that were created, modified, or used during an activity, and it defines an area of responsibility [18]. Normally, it is an output of an activity or a document used during a process.	*game design document*, *incremental build*, or *assets*.
Activities	Activities, tasks or steps describe units of work that provide a result. They can be *explicit* (explicitly declared) or *implicit* (inferred by the authors).	*programming*, *prototyping*, and *design*.
Practices	Practices are patterns or systematic habits used by teams during a development cycle. They can be associated with an activity.	*TDD*, *collaborative development* and *cutting features*.
Activity gateways	Activity gateways specify moments from start and end of parallel activities.	
Association	Association arrow are used when some elements are associated to one. For example, to make a comment about something in the process.	
Comments	Comments are additional information about some element or event.	

[3] All 20 postmortem process models are available on http://gas2016.github.io

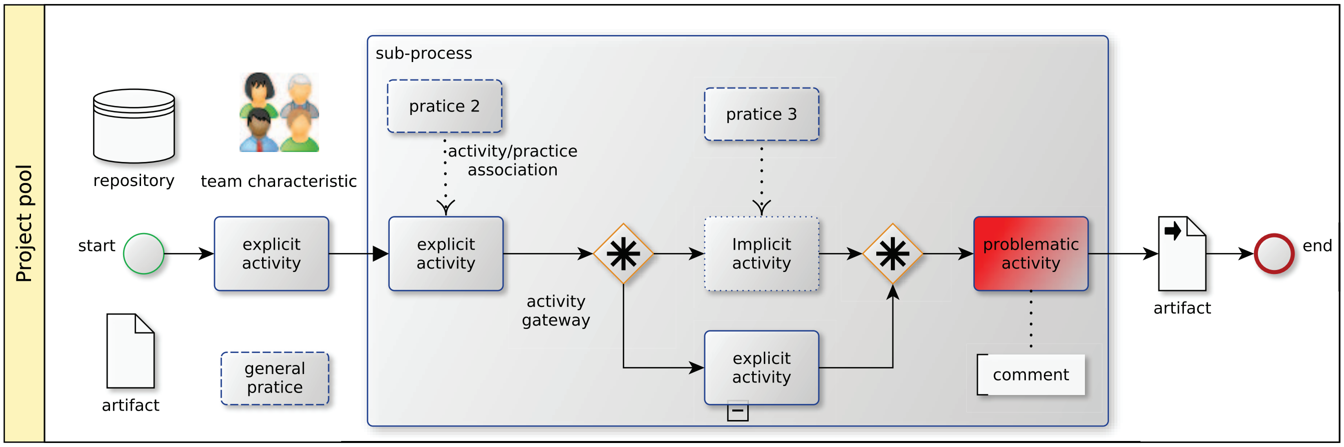

Figure 2: BPMN meta-model.

Table 3: Analyzed postmortem

Postmortem	Process	Agile
Brutal Legend	hybrid	yes
Kingdoms of Amalur: Reckoning	iterative	yes
Caseys Contraptions	iterative	yes
Sins of a Solar Empire	iterative	yes
Amnesia: A Machine for Pigs	iterative	yes
City Conquest	iterative	yes
Baldurs Gate Enhanced Edition	iterative	yes
Trine	waterfall	no
Natural Selection 2	iterative	yes
The Path	iterative	no
Dust An Elysian Tail	waterfall	no
Anomaly Warzone Earth	iterative	no
A Reckless Disregard for Gravity	ad-hoc	no
Scooby-Doo First Frights	waterfall	no
Spider-Man	hybrid	yes
Deadliest Warrior	waterfall	no
Zack Zero	waterfall	no
God of War Ascension	iterative	no
Electronic Symphony	waterfall	no
Guacamelee	iterative	no

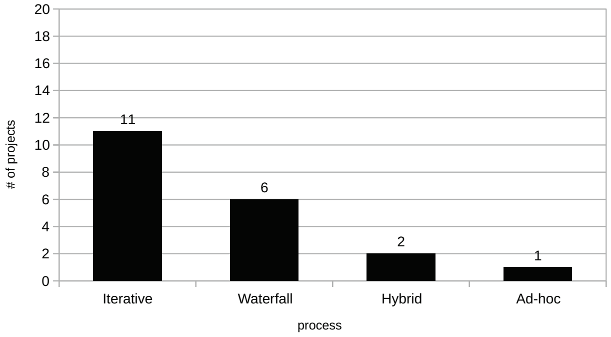

Figure 3: Process occurrences by category

The pre-production is composed by high concepts of the game, pitch construction, and milestones definition. Once this phase finished, development comes to the iterative production phase, where a set of activities are performed repeatedly, delivering a playable and testable game build in every iteration. When the project is nearly to be done, the team change for the waterfall approach, producing the remaining features and integrating all parts to deliver a gold version. Some agile practices are typically used like collaborative development, automated testing and continuous delivery.

Iterative process is defined by development cycles. *Kingdoms of Amalur: Reckoning* is a typical example of this process, modeled in Figure 5. The pre-production phase is usually short, producing a game main vision and decisions, followed by several complete cycles of development. During iterations, agile methodologies and practices (like Scrum [16]) are applied by 45% of analyzed projects. Some significant activities are sprint planning events, continuous delivering, and testing.

Ad-hoc process is exemplified in Figure 6. In this case, the development start with brainstorming and initial planning activities with use of game design document. The production can be separated by level creation, experimentation and testing. Tasks are distributed by role, following a contextual sequence. This approach is usually adopted by small teams.

Waterfall process consists of well-separated sequence of phases, showing in Figure 7. First, a game conception is made during the pre-production phase. Second, a complete game design is create during the design phase. Third, using this design, a game is implemented (programming, audio, art). There are not intermediate deliverables or scheduled milestones. Next, the game made is tested. Finally, the complete product is delivered. Waterfall process has usually well-defined multi-functional teams.

4.2 Result Analyzes

Analyzing these models and results, we conclude that **the "old days" are gone, but not completely**. Despite some particularities discussed by Murphy-Hill *et al.* [23], **this work shows that actually video game and traditional software development share similar processes and practices**. The video game industry has improved its processes, adopting regular software engineering techniques. In the same direction of previous academic studies, **we found that iterative process is actually *mainstream* in video game industry and agile practices adoption is increasing in the last years**. However, we believe that iterative process and agile practice benefits are

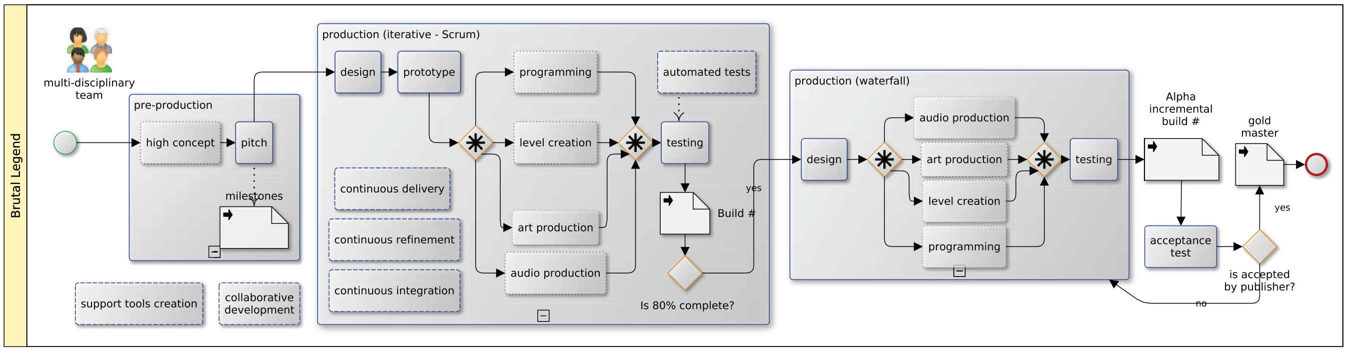

Figure 4: Hydrid process - Brutal Legend model

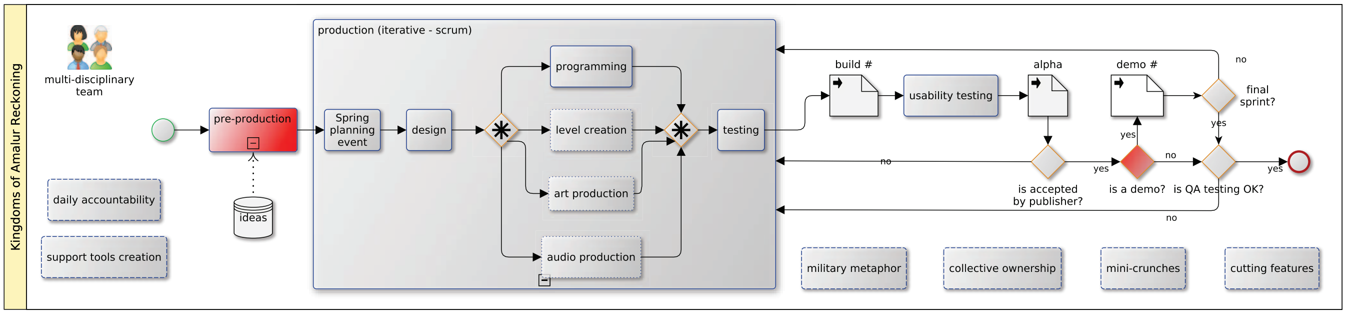

Figure 5: Iterative process - Kingdoms of Amalur: Reckoning model

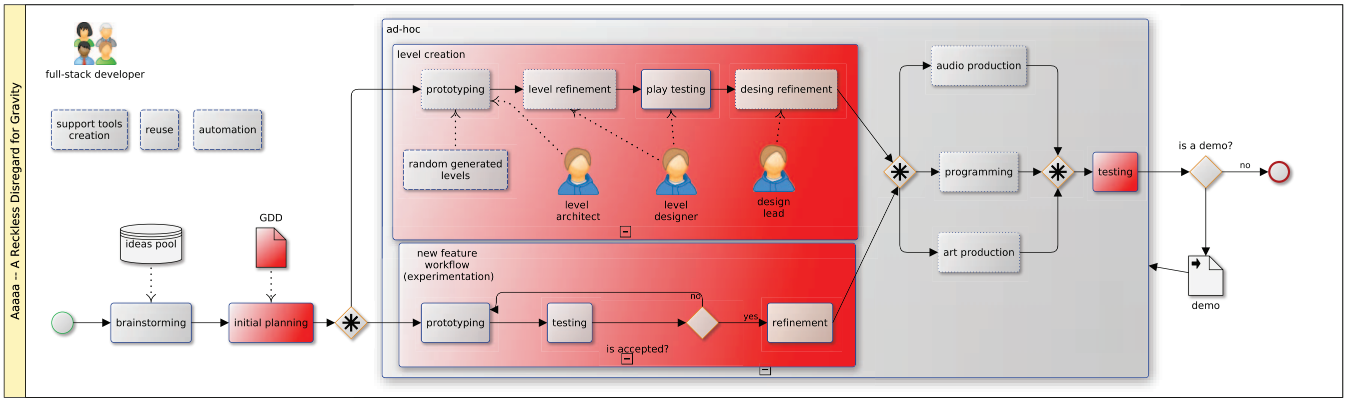

Figure 6: Ad-hoc process: Aaaa! A Reckless Disregard for Gravity model

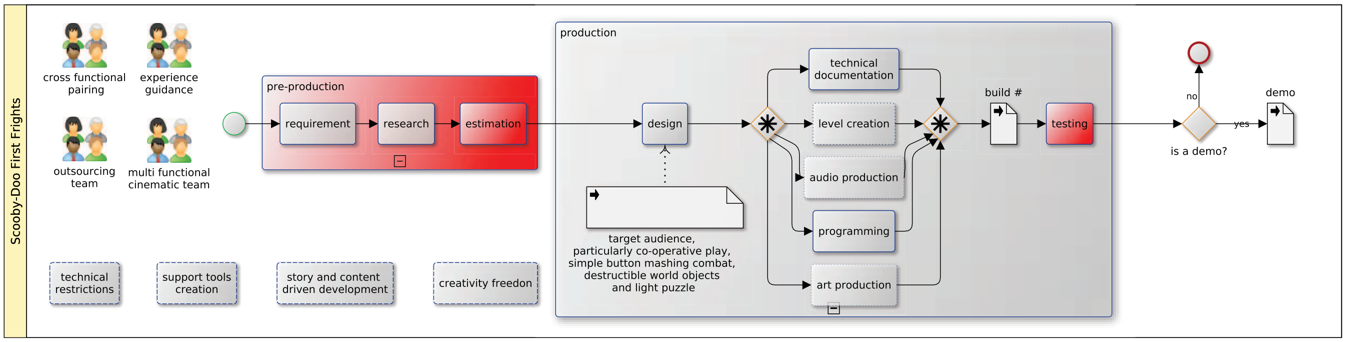

Figure 7: Waterfall process: Scooby-Doo model.

yet **misunderstood** by some game developers, managers, producers, publishers, and educators.

Furthermore, significant facts were found on *publishers* and *milestones*. Once a company (including indie) made a deal with a publisher, normally a set of milestones are predefined, and when a portion of the game must be presented. These milestones are usually underestimated, It can, most of the time, usually . This situation was observed in *Anomaly Warzone Earth* postmortem:

> *"Knowing from the past that chasing milestones could leave us with heavy crunch time, which was something that no one wanted at all, we made a production plan with solid buffers after each milestone, just in case we would slip."* - ***11 Bit Studios*** *"Anomaly Warzone Earth" Postmortem*

4.3 Limitations

There are some limitations in this work. First, it's hard to generalize with a sample of 20 postmortems but, nevertheless, it shows good clues and a starting point. Second, all the postmortems are gathered only from Gamasutra portal where all related games were successful delivered. Finally, as pointed by Washburn *et al.* [14], some postmortems' authors may be hided the true information.

5. RELATED WORK

Several researchers studied game development processes in the past years. O'Hagan *et al.* recently published two studies. First, they produced a Systematic Literature Review (SLR) of the software processes used in game development [27]. They analyzed 404 papers (73% of studies were nonindutrial), extracting 23 process models. They found that 47% of these models are agile and 53% are hybrid processes. In addition, they concluded that there is not one best model for game development. In their next work, they [25] investigated software process impacts on the game development, in a case study using Scrum methodology. More than conclusions, this work discusses several research opportunities. In many aspects, our survey is complementary to these studies because we analyzed 20 industrial projects and used a different source (postmortems) to get our research data.

In another recent study, Murphy-Hill *et al.* [23] interviewed 14 developers and collected 364 survey responses about game development activities. There results suggest that games have significant differences from traditional software development. They defend that game development process are not homogeneous, but instead are a rich tapestry of varying practices involving diverse people across domains.

Callele *et al.* [6] summarized requirements for game development and highlighted important research opportunities in this topic. Petrillo *et al.* [28] conducted a postmortem analysis study to identify good practices adopted in several game development projects, claiming that a deployment of agile methods like Scrum and XP can occur naturally, because teams already apply several agile principles in their activities. Our actual study shows that this result is valid, but agile practices are not smoothly applied as we believed at the time. Finally, Washburn *et al.* [14] analyzed 155 postmortems from 1998 till 2015 using a set of categories to highlight the best practices ("what went right") and the pitfalls ("what went wrong") occurred during the game development. Still, they offer recommendations to game developers.

Acknowledgment

We give our thanks to the postmortems' authors for sharing their cases. This work has been partly supported by the Natural Sciences and Engineering Research Council of Canada and the Canada Research Chair on Patterns in Mixed-language Systems, and by National Council for Scientific and Technological Development of Brazil.

6. CONCLUSIONS

In this paper, we presented a survey of software engineering processes in video game industry from postmortem project analyses. We analyzed 20 postmortems from Gamasutra Portal and extracted their processes. As a result, we found that **iterative practices** are increasing their adoption and are applied in at least **55% of projects**. Moreover, **agile practices** are explicitly adopted in **45% of projects**. **Waterfall** process is still applied in **30% of the projects**.

These results have several conclusions and implications. First, **the "old days" are gone, but not completely**. Despite some particularities, **this work shows that video game and traditional software development share similar processes and practices**. The video game industry has improved its processes, adopting software engineering techniques. In the same direction of previous academic studies, **we found that iterative processes are actually *mainstream* in video game industry and agile practices adoption is increasing in the last years**. However, we believe that iterative process and agile practice benefits are yet **misunderstood** by some game developers, managers, producers, publishers, and educators. To address this issue, **we suggest that educators include game agile methodologies in their regular game development courses**.

In future work, we plan two studies. First, we plan to evaluate if there is a correlation between actual project challenges and the processes and practices discussed in this work. Secondly, we plan to survey the postmortem authors, discussing about their projects to get more details and to evaluate our models.

References

[1] R. Al-azawi, A. Ayesh, and M. A. Obaidy. Towards Agent-based Agile approach for Game Development Methodology. In *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, pages 1–6. IEEE, jan 2014.

[2] K. Beck. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000.

[3] A. Birk, T. Dingsoyr, and T. Stalhane. Postmortem: never leave a project without it. *IEEE Software*, 19, may 2002.

[4] S. Breakfast. Scrum glossary: 62 scrum related terms in 50 words or less (each), 2016. [Online; accessed 19-January-2016].

[5] D. Callele, E. Neufeld, and K. Schneider. Requirements engineering and the creative process in the video game industry. In *13th IEEE International Conference on Requirements Engineering*, August 2005.

[6] D. Callele, E. Neufeld, and K. Schneider. A report on select research opportunities in requirements engineering for videogame development. In *2011 Fourth International Workshop on Multimedia and Enjoyable Requirements Engineering (MERE'11)*, pages 26–33. IEEE, aug 2011.

[7] M. Cohn. *Succeeding with agile: software development using Scrum*. Pearson Education, 2010.

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

[9] M. Fridley. Postmortem: Kingdoms of Amalur: Reckoning, 2013.

[10] 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.

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

[12] Ieee. *Guide to the software engineering body of knowledge version 3.0.* 2014.

[13] Innolution. Agile glossary definitions, 2016. [Online; accessed 19-January-2016].

[14] 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. ACM – Association for Computing Machinery, May 2016.

[15] C. M. Kanode and H. M. Haddad. Software Engineering Challenges in Game Development. In *2009 Sixth International Conference on Information Technology: New Generations*, pages 260–265. IEEE, 2009.

[16] C. Keith. *Agile Game Development with Scrum*. Addison-Wesley Professional, 1st edition, 2010.

[17] P. Kruchten. *The Rational Unified Process: An Introduction, Second Edition*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 2000.

[18] P. Kruchten. *The rational unified process: an introduction*. Addison-Wesley Professional, 2004.

[19] A. L. Labs. The agile dictionary, 2016. [Online; accessed 19-January-2016].

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

[21] R. C. Martin. *Agile software development: principles, patterns, and practices*. Prentice Hall PTR, 2003.

[22] R. C. Martin. *Clean code: a handbook of agile software craftsmanship*. Pearson Education, 2009.

[23] 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*, pages 1–11, New York, New York, USA, 2014. ACM Press.

[24] M. Myllyaho, O. Salo, J. Kääriäinen, J. Hyysalo, and J. Koskela. A review of small and large post-mortem analysis methods. In *IEEE France*, Paris, November 2004. 17th International Conference Software & Systems Engineering and their Applications.

[25] A. O. O'Hagan and R. V. O'Connor. Towards an Understanding of Game Software Development Processes: A Case Study. In D. Winkler, R. V. O'Connor, and R. Messnarz, editors, *Communications in Computer and Information Science*, volume 301 of *Communications in Computer and Information Science*, pages 3–16. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015.

[26] O. M. G. Omg, R. Parida, and S. Mahapatra. Business Process Model and Notation (BPMN) Version 2.0. Technical Report January, 2011.

[27] A. O. O'Hagan, G. Coleman, and R. V. O'Connor. Software development processes for games: a systematic literature review. *Systems, Software and Services Process Improvement*, pages 182–193, 2014.

[28] 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 - SIGDOC '10*, page 9, New York, New York, USA, 2010. ACM Press.

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

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

[31] Scrum.org. Professional scrum developer glossary, 2016. [Online; accessed 19-January-2016].

[32] A. September. Ieee standard glossary of software engineering terminology. *Office*, 121990(1):1, 1990.

[33] J. Shore et al. *The art of agile development.* " O'Reilly Media, Inc.", 2007.

[34] SolutionsIQ. Agile glossary, 2016. [Online; accessed 19-January-2016].

[35] Telerik. Agile terms glossary, 2016. [Online; accessed 19-January-2016].