Modelagem de Tarefas com Foco no Ator para Desenvolvimento de Jogos Indies: Uma Metodologia para Reter Conhecimento em Processos de Inovação

Actor-Focused Task Modeling for Indie Game Development: A Methodology to retain knowledge on innovation processes

Bruna Salles Reis[*]

Eduardo de Oliveira Sierra[**]

Resumo

Desenvolvedores de jogos indies representam uma considerável parcela dos recursos humanos da indústria do entretenimento. Enquanto empresas grandes, compostas por dezenas e até centenas de desenvolvedores, possuem processos de gerência de processos de inovação bem definidos, os pequenos grupos não os possuem. Este artigo propõe uma metodologia para reter conhecimento obtido da execução de tarefas, permitindo uma análise de post-mortem mais detalhada enquanto melhora o acompanhamento de tarefas em níveis pessoais e gerenciais. A ideia é conseguir mitigar os efeitos negativos provenientes da escassez de dados gerada pelo uso de métodos ágeis em empresas de jogos indies.

Palavras-chave: Desenvolvimento de Jogos, Engenharia de Software, Gerência de Projetos, Métodos Ágeis.

Abstract

Indie game developers represent a considerable amount of human resource from the entertainment industry. While big companies, composed of dozens or even hundreds of developers have well-established processes for managing innovation processes, small teams lack from it. This paper proposes a methodology to retain knowledge obtained from the task executions, allowing a more detailed post-mortem analysis while also enhancing the task management at both personal and managerial level. The idea is to mitigate the downsides of the data outage caused by the use of agile methods in indie game companies.

Keywords-component; Game Development, Software Engineering, Project Management, Agile Methods.

Aprovado em: 18/07/2019 Versão Final em: 22/07/2019

Trabalho de conclusão de curso apresentado ao curso de Bacharelado em Sistemas de Informação da Universidade Federal Fluminense como requisito parcial para conclusão do curso.

[*] Graduando do Curso de Sistemas de Informação-UFF; brunasallesreis@id.uff.br

[**] Graduando do Curso de Sistemas de Informação-UFF; eduardosierra@id.uff.br

1 INTRODUÇÃO

The challenges faced by the video game industry are very well known. The multidisciplinary nature of digital games which leads to multidisciplinary team management, the constant technology advancement and update that requires a constant innovative environment on the development process and the highly subjective concept known as fun are only examples of such challenges (Kanode and Haddad, 2009). These issues may be considered critical when combined in one project and lead to management problems being the vast majority of game development, as described by Petrillo (Petrillo, 2008). Even so, the industry became one of the most important and profitable creative sectors, generating $100 billion per year (Granados, 2018).

In the midst of such scenario, indie game companies, companies that are independent on a financial, creative and publishing basis (Garda and Grabarczyk, 2016), and that are normally composed by small development teams, face even more challenges, as the compact team composition often leads to accumulation of roles (Stacey and Nandhakumar, 2007), leading to a higher impact on every composition change and aggravating the already existent management issue.

Even though the video game industry has adopted the use of agile methods on a common basis, the simple use of these methods will not solve all the present management issues (Godoy and Barbosa, 2010), and their use might lead to another group of problems, such as data outage in the decision making and the development processes. The Agile Manifesto (Beck, 2001) defines that focus must be on the product, rather on the process itself, avoiding unnecessary documentation and giving the flexibility to accept and react to changes (Godoy and Barbosa, 2010). The informality and fast prototyping results coming from this strategy made this process a common and preferred approach to the game industry. However, such an approach might constantly cause a loss of valuable information on the ongoing innovation process and on the reasoning behind the changes.

This information could lead to a better understanding of the development context. Not only that, but the lack of documentation can also become a problem when communicating the requisites and changes through a distributed geographic team and on large projects (Inayat, Salim, Marczak, Daneva and Shamshirband, 2014).

Commonly, on indie game companies, where the high frequency of team turnovers causes the loss of valuable technical knowledge, which is critical to the knowledge achieved

on the execution of the innovation process. This is critical in order to secure the company maintainability and allow a better analysis to define the team reality and particularities to propose better fit methodologies.

While the use of agile methods may help the project management issues that are common in the game development scenario, it is widely agreed that most of them must be adapted to team reality. Moreover, to the particularities of game development, and even some proposed agile-based methodologies such as eXtreme Game Development (Demachy, 2010) and Game Unified Process (Flood, 2003) end up falling short in meeting the entirety of the management issues (Godoy and Barbosa, 2010).

Kanode and Haddad also discuss that it might be necessary to use different methods in different development phases (Kanode and Haddad, 2009), as each development phase has its own specific needs (Petrillo, 2008). Although, the literature lacks of methodology proposals for that aspect.

Our objective is to propose a methodology to retain the knowledge obtained from task executions in order to mitigate the downsides of the data outage caused by the use of agile methods.

2 DESENVOLVIMENTO

In section 2.1, we will explain how we thought the process defined in the methodology, our references and our expectations towards the result. In section 2.2, we will define the methodology process. In section 2.3, we will explain our experimentation methodology, defining the focus of our research and our goals. In section 2.4, we will present the results and compare them with our initial expectations. In section 2.5, we will discuss the written feedbacks sent by the research subjects and the ideas we had after them.

2.1 Methodology Conception

In order to mitigate the data outage caused by the use of agile methods, we decided to take a step towards the person in charge of executing the task itself, as innovation processes in game development require the freedom of choice of the best solution to be handed to the one that is executing the task (Garda and Grabarczyk, 2016). If we could allow the actor himself to freely describe and model his own task process, updating it accordingly as changes in the task happen, the diagram created by this method could contain a detailed path on how to solve

the modeled task, which obstacles appeared during the execution of the task and which decisions were taken during the execution, in a more detailed task log update.

This data could not only be used in the post-mortem analysis, but also to teach the path taken by the actor to other actors who did not take part on the task itself, or even use the path taken to support optimization of the task solution. Also, we believe that the simple act of modeling the task, when facing a task that is not known to the actor, might be helpful in terms of understanding the solution and identifying potential scope issues before starting the execution of each task.

From the first point of conception of the method, the modeling approach we propose is to match the idea with the Business Process Modeling System (BPMS), as it is already a methodology focused on the process in a generic aspect. Although, if we were to take the traditional BPMS approach, it would be complicated to get every actor to fully understand and execute its concepts, especially taking in consideration that many of those actors are not necessarily related to programming nor administration in general.

Thus, we decided to simplify most of the modeling process: First, we kept the step definition, but without separating it in actor pools, as the only actor is already modeling his own steps, without detailing the artifact references; Second, we decided to keep the sequence demonstration through the use of arrows, going from a step to another. Although, lifting the constraints of their use, therefore allowing steps without sequences and steps that could lead to multiple other steps, to ensure the freedom of the process. To ensure a better tracking of the task progression, we also decided to add a virtual Kanban representation through three step status, indicated in the steps as empty circles. The amount of colored circles indicates the current step status, being: none, not started; one, ongoing work; two, waiting for validation; and three, completed. In order to provide a more precise task progress representation, the actor should consider the status of all steps listed in the task and their relative weights, if any were defined.

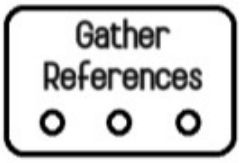
While the visual representation could map the task process by itself, the inclusion of a written changelog was defined as vital in order to store the reasoning behind the changes of paths during the execution. The simple visual representation of the paths would not hold the description of the obstacle or reasoning behind the decision and, even if we were to create a group of symbols to address a type of obstacle, it would still lose much information that is critical about the decision taking, which is one of the main objectives of this methodology. Through both task process mapping and change log, we believe this methodology could be

effective on data gathering while also providing better support to task management and overview.

2.2 Actor-Focused Task Modeling

The actor-focused task modeling process begins when the actor receives a new task. As soon as he is briefed about the task, he must split the task into the steps he believes are necessary for the conclusion of the task. Those steps must be clear and relevant, in order to avoid over detailing. If the conclusion path is not yet clear for the actor, as he may have to find a way to complete the task as he executes it, he must model the steps that are already clear for him and update the model accordingly to his progress over finding the solution as a change of path. The steps must be drawn as rectangle boxes, with the task name and three circles inside it (Figure 1). If there is any sequential dependency of the steps, the dependency must be represented through the use of a directional arrow, pointing the next step in the sequence. A step may have multiple dependencies, and it shall be represented by having more than one arrow pointing to the dependent step.

Figure 1. Example of a step.

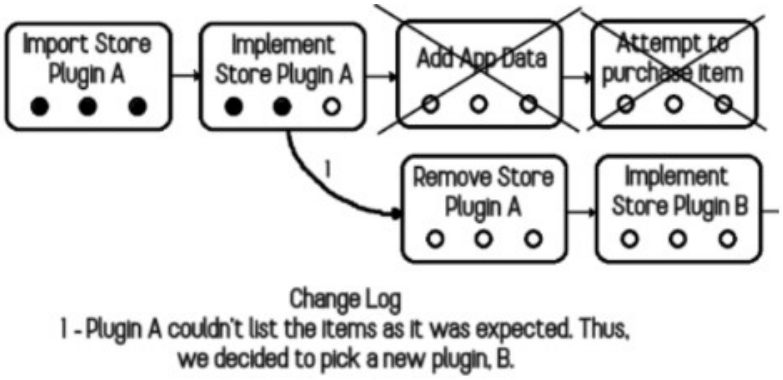


Fonte: Dados da pesquisa.

The model must be updated whenever a step is concluded, at the end of every workday and at every change of path on the task execution. The first two cases will be represented by the coloring of the circles inside the steps, while the latter will imply a new modeling phase. A change of path may be caused by briefing changes and misunderstandings, unforeseen technical issues, and other reasons. When a change of path happen on the task execution, the motive of the change must be listed under the Change Log and have an identifier number attached to it. After that, all the steps that won't be executed anymore, as any tasks that had any dependency on them, must be marked with a cross, indicating they won't be executed anymore. Then, the actor must model the new path and the further steps required to complete

it. The first step of the new path must be indicated by a directional arrow from the last executed step, together with the change motivation identifier (Figure 2).

Figure 2. Example of an updated model after a path change.



Fonte: Dados da pesquisa.

The model created by the methodology should be stored by the company. If an actor is doing a task that has already been modeled, the previous model should be used and updated, instead of remodeling the task. In the case of a different path is proposed, a new model must be made and, if it fails, it should be attached to a successful model as a new change of path, having all steps already crossed. If the new path is successful, it should be kept as a different model related to the same task.

2.3 Experimentation Methodology

As it would be quite challenging to convince a company to use the methodology throughout an entire project, due the undefined amount of time it would require, we decided to get developers from the Brazilian indie game development scenario to try using the methodology during a single task. This would answer if the methodology objectives on retaining knowledge and assisting the task management would be happening and if the members of the industry consider both the problem and this solution proposal relevant at all. For that, we divided the questions into two distinct forms and sent both of them to two indie game development groups, one for companies located in Rio de Janeiro and one for companies located in São Paulo.

The first form is a profile check and should also work as a briefing for the use of the methodology, inviting the subject to join the experiment. In this form, we intended to validate if the subject's company was fitting the indie company concept through the amount of employees, the agile method dissemination in the area, Godoy and Barbosa's claim on their

frequent modification in real application (Godoy and Barbosa, 2010) and Stacey and Nandhakumar's claim on role accumulation in game development companies (Stacey and Nandhakumar, 2007).

The second form is a directed feedback of the single task execution of the methodology. We defined a group of eleven questions to be answered in a 5-points Likert scale, focusing on their experience and how they felt during the execution. Before focusing on the effectiveness of the methodology quantitatively, we wanted to make sure that it was easy to use and considered relevant to the industry. One of our main concerns was the possibility that the methodology was too intrusive as it relates to the actors own knowledge. For some of them, this might be seen as a threat as it could lead to a loss of importance on the development process or even be considered a path to a forced process change, going against the freedom of choice on the task execution. To address this concern, we added a question about the subject's opinion on having their diagram observed by others and a second field with a couple of expected reasons for uneasiness on such observation, also allowing the subject to add other reasons to the list. As a last question, we asked for critics and suggestions on the methodology itself.

2.4 Results

The experimentation was done with a total of 19 subjects on the first form, from which 13 tested the methodology and gave feedback afterward on the second form.

Table 1. Number of simultaneous agile methods use.

Number of Agile Methods Used	Subjects
0 or Unknown	4
1 Method	6
2 Methods	7
3+ Method	2

Fonte: Dados da pesquisa.

Table 2. Number of simultaneous occupation areas.

Number of Occupation Areas	Subjects
1 Area	10
2 Areas	6
3 Areas	2
4+ Areas	1

Fonte: Dados da pesquisa.

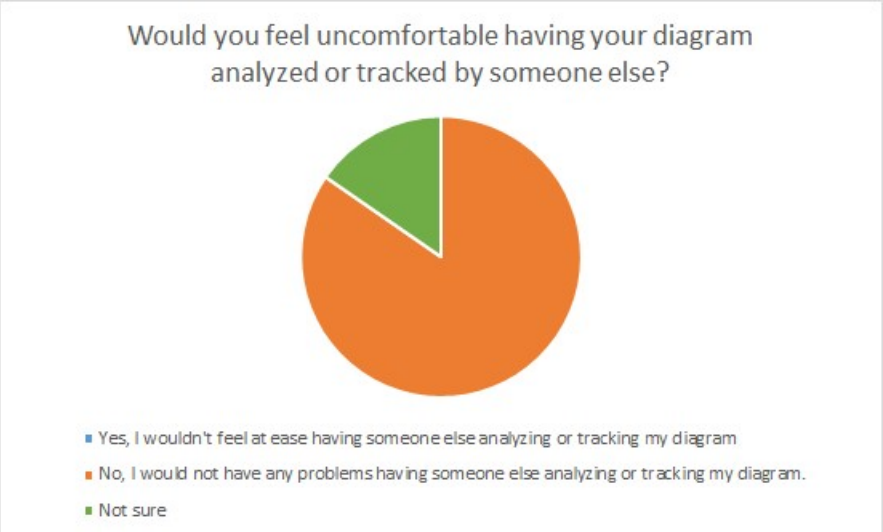
In the first form, the first point to be noted was the size of the teams, which matched what we were expecting on the definition of indie game company, as 94,7% of the participants were in teams up to 12 members. Following that, the dissemination of the agile methods on the industry (Table 1), only 10,5% were certain of not using any agile methodology and another 10,5% were uncertain about it. That implies that 79% of the work group uses agile methods effectively. Of those, SCRUM was the most used, representing 93,3% of agile users, and was most frequently combined with other methodologies, especially with Kanban, that appeared in every reported combination. About the agile method modification, the lower ratio of modification found of all the listed methods was 78,5% on SCRUM. Overall, there is a 81,2% modification rate for agile methodology usage. This result reaffirms Godoy and Barbosa's claim on the frequency of agile methods modification. Regarding Stacey and Nandhakumar's claim on role accumulation, 9 out of the 19 subjects had some area related accumulation (Table 2), where the most common occurrence was programming with some level of management responsibility, at 66,6% basis.

In the second form, most of the answers on the Likert scale followed our expectations. Most of the testers did not feel the methodology was interfering much with the other methods used and felt that, at some point, it complemented the other methods. In the knowledge retainment, while most of the testers agreed that checking their own diagram from a year before would help them remember their situation and mindset at some point, they were highly positive that using someone else's solution of a task would help them executing a similar task. They also agreed that the information held by this methodology is important to the company. For the task management topic, most of the answers were positive that the methodology helped to have a better view and understanding of the task to be executed and that the diagram helped them keep a better focus and a better track of the task development. Moreover, in the topic of the demand for methodologies to address the downsides of the agile methods and if the proposed methodology was able to address some of those downsides, the answers are mixed between partially agreed and unsure. This result could indicate that those downsides are not completely clear to most users and, thus, it might be hard to properly address them without a throughout explanation of both the agile method context and their bright and down sides.

To our surprise, none of the subjects showed any negative reaction towards having their diagrams analyzed or accompanied by others (Figure 3). While 15,4% did not know how they would react, the other 84,6% answered that they would not have any problem towards that. In an additional note, one of the subjects marked that a reason for not feeling

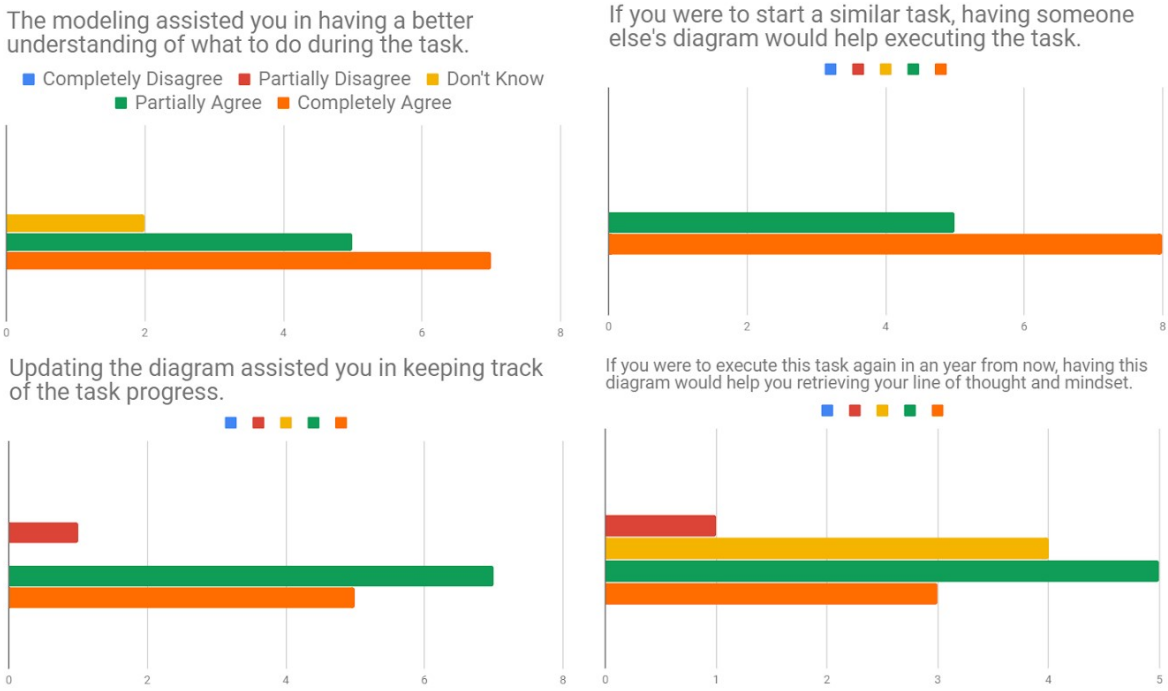
comfortable having their diagram analyzed or accompanied by others was that the subject feared that the modeling would end up molding or modifying their way of executing tasks, even though the subject didn't feel that they were uncomfortable towards the influence of others. This raised issue relates to our initial concern of a possible uneasiness while executing the methodology but, as observed in the overall result, it does not seem to be a critical matter as we thought it could be.

Figure 3. Results for the uneasiness question on the second form.



Fonte: Dados da pesquisa.

Figure 4. Likert scale answers to some of the questions of the second form.



Fonte: Dados da pesquisa.

Other than the expected uneasiness not appearing in the results, most of the results were within our expectations and showed that there is a relevant gap to be filled in this matter.

2.5 Discussion

From all 13 answers for the second form, 7 of them added a more detailed feedback over their experience with the methodology. In order to approach the content of those feedbacks, the content was separated in topics: tutorial, overhead and similar methods.

Some answers addressed issues with fully understanding of the created tutorial created to present the methodology. The diagrams used to illustrate the methodology contained steps such as “testing”, even though the step status already includes a validation status, making the “testing” step useless. Not only that, the step status did not have a good explanation of their represented moment nor there was a good definition on how to delimit the steps division, as its left to user’s behalf. Both the exemplified diagram and the step status explanation are candidates for change, as they are clearly negatively influencing the understanding of the methodology explanation. Although, the step division delimitation might be a more complicated matter, since the methodology has no intention of forcing a limit on the degree of detailing on the diagram. However, taking an extreme approach on the level of detail would most likely undermine the methodology effectiveness. In order to keep the freedom of choice over the step division delimitation, it was decided to add a suggestion on the level of detail and warn about the effects of lowering and expanding this level of detail.

On the overhead topic, one of the subjects made sure to comment that, in his opinion, this methodology would fit in his definition of excessive investment on documentation and would not be worth on small self-managing teams and projects, as this time could be better invested in better code structure and commenting. Sometimes it takes more time to document than to execute the task, he quoted. On this matter, the methodology time consumption should not take longer than the task execution and, once modeled or when using another task’s diagram as a reference, the time used for designing should go exponentially down. The effectiveness of the methodology on self-managing teams still seems relevant as the maturity of those teams is highly dependant on the team composition (Wagner III and Hollenbeck, 2012) and, as any composition or role change is made, the knowledge over the existing roles should be preserved to mitigate the negative effects of the changes. As proved by the results over knowledge retainment, the use of another’s diagram should most probably assist the

execution of the tasks by someone who is not used to the task itself. Thus, as the preservation of knowledge is, at some point, assured by the use of the methodology, self-managing teams would be able to deal with changes in a more flexible way.

As for similar methods, while one of the subjects mentioned a certain similarity with MeisterTask, another commented that his company had adopted a modified version of the methodology to their documentation and management tool, called Notion. In the MeisterTask comparison, the main differences were that MeisterTask does not keep track of the steps that turned obsolete after a change, does not keep any records about the reasons of change and it does not display a sequential order in a visual representation. The subject's mention referred to the actor-focused task modeling as a "better version" and a "great analysis tool". In the company adaptation of the methodology, the visual aspect of the methodology was ignored and replaced by a text-based approach, which he said to fit their demands.

Overall, the feedback of the methodology itself was quite positive, even though some changes in the methodology may be proposed.

3 CONCLUSÃO

The proposed methodology was considered successful in meeting the expectations pointed in this paper. The methodology did not create an aggressive environment for the actor's development process and had a positive response on its use. Although, it is also understood that it requires further increments in order to become a viable and reliable methodology.

Since our proposal had some considerable level of acceptance on the context of innovation process preservation, a more detailed analysis over the methodology effectiveness and it actual processual cost is what we believe that would be the next step towards enhancing this tool on our future work.

Also in our future work, we believe it would be interesting to take a better look at a modular approach on agile methods, in order to create an environment of simpler agile methods that could be easily grouped together to fit the team reality and his development needs. Instead of forcing each development group to adopt and modify a complex and rule heavy methodology such as SCRUM or RUP, it could be better to use simple, objective and independent smaller methods that could be combined and removed to fit the company environment better. At this point, it is quite hard to propose changes in the current agile methods as they are so constantly personally adapted. If a common basis for those methods

could be created, it would be possible to identify and properly address the issues that cause the constant adaptations, thus helping to establish a more reliable development scenario for game development.

REFERÊNCIAS

BECK, BEEDLE K., VAN BENNEKUM M., COCKBURN A., CUNNINGHAM A., FOWLER W., GRENNING M., HIGHSMITH j., HUNT J., JEFFRIES A., KERN R., MARICK J., MARTIN B., MELLOR R.C., D. 2001, "**Agile Manifesto**" , October 2009

DEMACHY, T., 2003. "**Extreme game development: Right on time, every time**", July 2010.

FLOOD K., 2003. "**Game unified process**", ARTICLE1940.ASP., July 2010.

GARDA, M. B and GRABARCZYK ,P., "**Is Every Indie Game Independent? Towards the Concept of Independent Game**", Game Studies, October 2016, Volume 16, Issue 1, ISSN:1604-7982

GODOY, A. and BARBOSA, E. F., "**Game-Scrum: An Approach to Agile Game Development**", Instituto de Ciências Matemáticas e de Computação (ICMC-USP), São Carlos (SP), Brazil, 2010, pp. 292-295

GRANADOS, C., "**Creative Industries and its innovation process: The case of mobile games**", Copenhagen Business School, Copenhagen, Denmark, June 2018, pp. 1-12

INAYAT I. , SALIM S.S., MARCZAK S., DANEVA M. and SHAMSHIRBAND S., "**A systematic literature review on agile requirements engineering practices and challenges**", 2014, pp. 2-13, doi: 10.1016/j.chb.2014.10.046

KANODE, C. M. , HADDAD, H. M. , "**Software Engineering Challenges in Game Development**", Sixth International Conference on Information Technology: New Generations, 2009, pp. 260-265, doi:10.1109/ITNG.2009.74

PETRILLO, F., PIMENTA, M., TRINDADE, F., and DIETRICH, C., "**Houston, we have a problem..: a survey of actual problems in computer games development**", Proceedings of the 2008 ACM Symposium on Applied Computing, Fortaleza, Ceará, Brazil, March 2008, pp. 707-711

STACEY, P. and NANDHAKUMAR, J., "**A temporal perspective of the computer game development process**", Warwick Business School, University of Warwick, Coventry, UK, 2007, pp. 479-497, doi: 10.1111/j.1365-2575.2007.00273.x

WAGNER III J.A., HOLLENBECK J.R., "**Comportamento Organizacional: Criando vantagem competitiva**", São Paulo, 2012, pp.192-203

APÊNDICE A – Lista de ilustrações

APÊNDICE B – Lista de tabelas