

Ludificação em Engenharia de Software: Tornando o processo de desenvolvimento em uma empresa um jogo

Gamification in Software Engineering: Making the development process of a company a game

Joel Coutinho da Silva Neto¹  <http://orcid.org/0000-0002-9020-401X>

Joabe Bezerra de Jesus Júnior¹  <http://orcid.org/0000-0002-1518-0572>

¹ Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil.

E-mail do autor principal: Joel Coutinho da Silva Neto joel.neto.90@gmail.com

Resumo

A importância de elementos de jogos no ambiente de trabalho, recentemente, tornou-se parte da prática de gestão emergente. Quando este processo é consentido, existe um aumento de efetividade nas atividades relacionadas ao trabalho. Desenvolver um software não é uma tarefa fácil, e as vezes, pode se tornar entediante devido à práticas de atividades constantes. Por esse motivo, o presente estudo transforma o processo de desenvolvimento de software de um ambiente corporativo em um jogo, aplicando técnicas de ludificação em uma equipe de desenvolvimento, inserindo os elementos de jogos em uma ferramenta de gerenciamento a qual a equipe usa para cadastramento de atividades.

Palavras-Chave: Jogo, Ludificação, Software, Desenvolvimento, Scrum, Empresa.

Abstract

The importance of gaming elements in the workplace has recently become part of the emerging management practice. When this process is consented to, there is an increase in effectiveness in work-related activities. Developing software is not an easy task, and sometimes, it can become tedious due to the practices of constant activities. Thus, the present study turns the process of software development from a corporate environment into a game, applying gamification techniques in a development team, inserting the games elements into a management tool that the team uses to register activities.

Key-words: Game, Gamification, Software, Development, Scrum, Company.

1 Introdução

A comunidade de jogos **online** (incluindo consoles, computadores e celulares) conta com milhares de indivíduos ao redor do globo. Várias pessoas estão escolhendo passar mais tempo em um mundo virtual do que no mundo real. Os jogos provêm recompensas, eles ensinam, inspiram e aproximam de uma maneira a qual a realidade não é capaz [1:4-5].

Na década de 1980, estudiosos como Thomas Malone, começaram a olhar para os games como fontes de heurísticas para interfaces agradáveis [2]. Nos anos 2000, o movimento dos chamados “jogos sérios” seguiu construindo jogos para educar, treinar e persuadir. Em paralelo a isso, campo de Interação Homem-Computador começou a explorar várias formas de tornar a experiência dos usuários mais agradável, divertida e motivadora [3].

As formas de entretenimento que os jogos digitais proporcionam, impactam nos comportamentos sociais e culturais de uma geração, o mesmo pode ser dito com relação as práticas profissionais [4]. A inserção de elementos de jogos no ambiente de trabalho, recentemente, tornou-se parte da prática de gestão emergente. Quando este processo é consentido existe um aumento de efetividade nas atividades relacionadas ao trabalho [5].

A aplicação de mecânicas de jogos em tarefas do mundo real, para influenciar o comportamento, aumentar a motivação e promover engajamento é denominada por Marczewski [6] como gamificação, ou ludificação.

Para Passos e colaboradores [7], o desafio de produzir um software raramente está relacionado a diversão. Na maioria das vezes, em equipes de desenvolvimento, requer uma hierarquia, aprender novas habilidades e muito trabalho em conjunto. Supreendentemente, isso é muito parecido com a definição abstrata de um jogo.

1.1 Objetivo Geral

O presente estudo tem como objetivo realizar um experimento, aplicando técnicas de ludificação em um ambiente corporativo em equipes de desenvolvimento de software durante o fluxo de implementação, com o intuito de promover o engajamento e aumentar o desempenho nas atividades.

1.2 Objetivos Específicos

- Coletar dados de um projeto de software que uma equipe desenvolveu ou que esteja desenvolvendo.
- Identificar os principais processos realizados pela equipe de desenvolvimento.

Definir quais elementos de jogos são mais apropriados para aumentar a motivação e o engajamento entre os integrantes da equipe, de acordo com suas atividades.

- Criar uma ferramenta para analisar os dados coletados, inserir os elementos de jogos e demonstração do resultado.

2 Referencial Teórico

Este capítulo descreve brevemente o conteúdo teórico utilizado como base para tentar resolver o problema descrito na seção 1.

2.1 Engenharia de Software

A Engenharia de Software é uma disciplina que leva em consideração todos os aspectos da produção de um software [8:1].

Em geral, engenheiros de software adotam uma abordagem sistemática e organizada para a realização do seu trabalho. Com isso, se tem mais efetividade na criação de um software de alta qualidade. Essa abordagem sistemática é chamada de processo de software [9:8].

Existem vários modelos que descrevem o fluxo de um processo de software, porém, o Rational Unified Process (RUP) tenta combinar a maioria dos elementos contidos nos modelos existentes em um único diagrama, agrupando as atividades em fases associadas a fluxos de trabalho. Cada fase representa o estado em que do software se encontra e os fluxos de trabalho são as atividades com maior ênfase naquela fase (vide Figura 1).

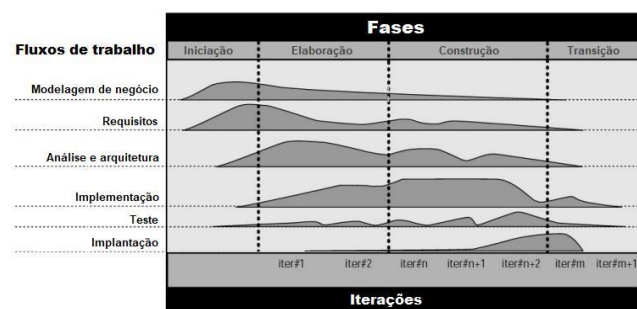


Figura 1 : Modelo de gráfico iterativo do RUP. Fonte: Imagem adaptada de [10: 3].

Na Engenharia de Software, a equipe de desenvolvimento é responsável pelo processo de codificação e tem suas atividades relacionadas ao fluxo de trabalho de implementação [10:12]. Esse fluxo tem uma ênfase maior nas fases de elaboração e construção, demonstrando um grande volume de participação no ciclo de vida do produto, como apresentado na Figura 1.

2.2 Desenvolvimento Ágil

Os negócios atualmente operam em uma escala de mudança global rápida. Por causa desse ambiente, é praticamente impossível trabalhar com uma sequência de requisitos estáveis. Os requisitos iniciais são alterados constantemente durante a produção do software, tornando difícil prever o impacto dessas mudanças nas práticas de trabalho e como o software vai se comportar [9:57].

No final da década de 1980, a necessidade de construção de software que se adapte a mudanças de requisitos constantes começa a ser reconhecida e o termo "desenvolvimento ágil" surge como uma metodologia que tem como base uma série de entregas incrementais [11]. Ou seja, parte do software é entregue a cada período, em vez de uma entrega única ao final de sua produção,

tornando os processos mais adaptáveis as mudanças constantes de requisitos.

2.3 Scrum

O Scrum é uma metodologia de desenvolvimento ágil que vem sendo utilizada desde a década de 90. Ele é definido como um framework de trabalho que engloba um conjunto de regras, papéis e eventos que devem ser seguidos [12:1].

Os papéis desempenhados dentro do Scrum são os de: Scrum Master, que está presente em todos os eventos e garante que a metodologia seja aplicada e entendida; o Product Owner, que é responsável por maximizar o valor do trabalho que o Time faz; e o Time, que executa o trabalho propriamente dito [13:4].

Os requisitos para o software que o Time está desenvolvendo são listados e chamados de backlogs do produto, geralmente representados como estórias de usuário (casos de uso). O Product Owner é o responsável pelo seu conteúdo, disponibilidade e priorização [12:16].

Como as metodologias ágeis têm como base entregas incrementais, no Scrum a sprint é um evento com duração fixa que consiste em quatro etapas:

1. Planejamento: Os backlogs do produto são apresentados, estudados e estimados, tornando-se backlogs da sprint.
2. Execução: A execução do trabalho que foi planejado. Pode durar, no máximo, até quatro semanas, dependendo do trabalho e complexidade do backlog da sprint. Durante essa etapa ocorrem as reuniões diárias, onde uma vez por dia, cada membro do Time explica o que fez, o que está fazendo e se existe algum impedimento.
3. Revisão: Apresentação e revisão do trabalho executado para o Product Owner.
4. Retrospectiva: Reunião de retrospectiva para troca e compartilhamento de experiências, com

a premissa de melhorar o desempenho do Time na próxima sprint.

Todo esse fluxo se repete até que o produto esteja completo (vide Figura 2) [12:10-15].

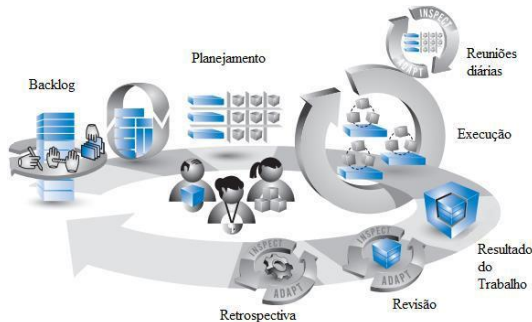


Figura 2. Fluxo de uma sprint. Fonte: Imagem adaptada de [13:17].

2.4 Ludificação

Originada na indústria de mídia digital, o termo gamificação ou ludificação é definido como o uso de elementos de jogos em um contexto, produto ou serviço que não seja um jogo [14]. Desse modo, inserindo elementos contidos em jogos em alguma atividade, produto ou serviço é possível torná-lo ou deixá-lo parecido com um jogo.

Segundo Yohannis e colaboradores [15], para identificar se um objeto está ludificado ainda são usadas duas abordagens diferentes, uma pelo significado léxico do termo, e a segunda através do ponto de vista de processo.

A partir do significado léxico, pode-se inferir ludificação como a ação de tornar algo em um jogo. Já o ponto de vista de processo, pode ser determinado como um fluxo onde um objeto com poucas características de jogo (estado inicial) é ludificado (ação), e como produto final, tem-se um objeto com mais identidade de jogo (estado final), portanto, houve o processo de ludificação (vide Figura 3).

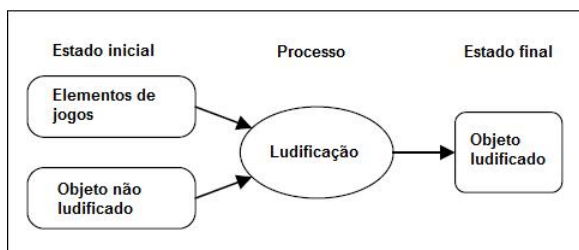


Figura 3. Ludificação do ponto de vista de processo. Fonte: Imagem adaptada de [15].

Apesar do termo ser bastante mencionado, ainda não é possível definir com precisão quais elementos estão dentro da lista de elementos de jogos – também conhecidos como mecanismos de jogos. Uma das soluções é tratar esses elementos como uma série de características em comum compartilhadas pelos jogos digitais [14]. Para Zichermann e Cunningham [16], algumas dessas características podem ser: quadros de liderança, medalhas, prêmios, pontuação, desafios e punições.

As áreas em que a ludificação pode ser utilizada são extensas, como a de saúde [17] e a de educação [18]. Ainda é possível aplicar a ludificação em todas as etapas da área de Engenharia de Software, onde mecanismos de jogos, mapeados de forma sistemática, podem ser incorporados em todos os fluxos do processo de software [19].

2.5 Ludificação em Engenharia de Software

De acordo com Pedreira e colaboradores [19], na Engenharia de Software, as áreas em que há maior interesse no campo da ludificação são as de requisitos, implementação e testes. O que, segundo eles, não é uma surpresa, visto que essas áreas compartilham atividades constantes e que podem chegar a ser tediosas, tornando-se assim alvos claros para aplicação da ludificação.

Os autores também relacionam os elementos de jogos mais utilizados e bem aceitos na área de Engenharia de Software (vide Figura 4). Dentre esses elementos, são destacados os mais expressivos:

- Sistema de pontuação: O jogador recebe pontos após completar alguma atividade.
- Emblemas: O jogador ganha uma titulação após atender algum critério.

- Classificação: Uma classificação da pontuação para promover competitividade.
- Premiação: São cumulativos e representam certas conquistas do jogador.
- Reputação: O jogador ganha fama, sendo reconhecido pela comunidade do jogo.

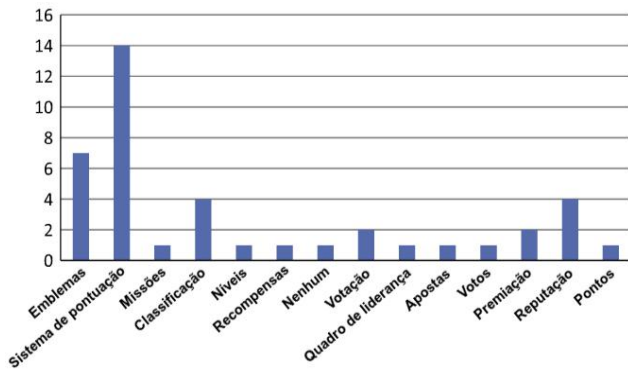


Figura 4. Elementos de jogos mais utilizados na Engenharia de Software. Imagem adaptada de [19].

Segundo Zichermann e Cunningham [16], existem dois elementos de jogos que estão relacionados e que, apesar de não estarem listados nos resultados das pesquisas dos mais utilizados, são de extrema importância por serem usados como uma técnica para manter o jogador motivado e engajado no jogo (vide Figura 5). São eles:

- Feedback imediato: O jogador é constantemente alertado sobre o seu progresso, conquistas e falhas.
- Ciclo de engajamento: Identificação do que pode levar o jogador a sair, e o mais importante, o que pode trazê-lo de volta para que continue engajado e motivado a jogar.



Figura 5. Ciclo de engajamento relacionado ao feedback imediato. Fonte: Imagem adaptada de [16:68].

Após implantar a ludificação em mais de cem empresas, Deterding [3] afirma que o processo de abordagem é sempre o mesmo. Começa com um entendimento do objetivo do negócio, depois é identificado quais atividades praticadas agregam valor, mesmo que indiretamente e em seguida um entendimento profundo dos usuários e o que os levam a se engajarem com esses objetivos.

Em um estudo realizado por Passos [7], é proposto um framework de trabalho baseado em desafios, punições, conquistas e feedback, onde as atividades completadas concedem pontos de valor ao desenvolvedor. Nesse modelo, uma ferramenta de gerenciamento de atividades é necessária para rastreamento de informações.

Cavalcanti [20] simula um experimento utilizando o framework de trabalho proposto por Passos e considera uma equipe de desenvolvimento que adota Scrum como metodologia ágil. Em seguida, são inseridos elementos de jogos em uma ferramenta de gerenciamento de atividades utilizada por essa mesma equipe, o Agilo For Trac [21]. O autor também cria um módulo que auxilia na análise dos dados, onde no final de cada ciclo, é apresentado ao desenvolvedor informações sobre o seu progresso e resultados do jogo.

3 Materiais e Métodos

Para realizar o experimento em uma equipe de desenvolvimento de software no ambiente corporativo, foi escolhida a empresa da área de tecnologia da informação e comunicação que será tratada como empresa P. Assim, esta seção descreve os materiais para o processo de ludificação que foi definido para a empresa e que é apresentado na seção 4.

3.1 A Empresa

Criada em 2005 pelo Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R.), a

<http://dx.doi.org/10.25286/rep.v2i4.578>

empresa P faz parte de um dos principais parques tecnológicos de inovação do Brasil, o Porto Digital [22], e conta com mais de 10 anos de mercado com experiência no desenvolvimento e consultoria em soluções inovadoras e complexas para clientes tanto do setor público, quanto do setor privado.

3.2 Seleção do Projeto de Software

O projeto de software (cujos dados foram disponibilizados pela empresa P) possui cerca de cinco mil registros de atividades, entre elas, tarefas e problemas cadastrados que foram implementados e resolvidos pela equipe de desenvolvimento. Qualquer descrição que contextualize sua finalidade foi mantida sob sigilo.

O software levou cerca de dois anos para ser finalizado e devido ao seu tamanho e complexidade foi dividido em seis iterações (vide Tabela 1). A cada período, a equipe implementava uma série de funcionalidades, resolvia problemas encontrados, e registrava suas atividades na ferramenta de gerenciamento. Em seguida, uma parte do software era entregue e uma nova iteração era iniciada.

Tabela 1. Iterações do projeto de software.

N ^o	Propósito	Utilizada neste experimento
1	Prototipagem	Não
2	Implementação e testes	Sim
3	Implementação e testes	Sim
4	Implementação e testes	Sim
5	Implementação e testes	Sim
6	Implementação e testes	Sim

Os dados relacionados a todas as iterações foram coletados, exceto os registros da iteração de número 1. A mesma se trata de protótipo do software e definição de arquitetura. Apenas três membros da equipe participaram e as atividades não estão relacionadas a nenhuma

implementação de funcionalidade. Por esse motivo, também não existem registros nos dados coletados. Essas circunstâncias inviabilizam, portanto, a implementação da ludificação e a análise de dados da primeira iteração.

3.3 Definição da Equipe de Desenvolvimento

Vários desenvolvedores participaram do projeto de software. Alguns deles, não permaneceram até a sua conclusão, por diversos motivos. Outros, foram integrados à equipe e começaram a participar em meio a fases já avançadas da implementação (vide Tabela 2).

Esses mesmo integrantes adotaram o Scrum como metodologia de desenvolvimento de software ágil e por uma questão de sigilo de informação, os mesmo não tiveram seus nomes divulgados e serão tratados neste estudo conforme consta na Tabela 2.

Tabela 2. Desenvolvedores do projeto de software.

N ^o	Desenvolvedor	Período participando
1	Desenvolvedor LA	Iteração 3 à 6
2	Desenvolvedor LE	Iteração 1 à 6
3	Desenvolvedor UO	Iteração 2 à 4
4	Desenvolvedor RU	Iteração 2 à 3
5	Desenvolvedor RE	Iteração 3 à 6
6	Desenvolvedor EN	Iteração 1 à 6
7	Desenvolvedor AA	Iteração 3 à 3
8	Desenvolvedor OE	Iteração 2 à 6
9	Desenvolvedor EA	Iteração 3 à 3
10	Desenvolvedor EU	Iteração 1 à 6
11	Desenvolvedor HO	Iteração 2 à 2
12	Desenvolvedor HU	Iteração 2 à 6
13	Desenvolvedor IA	Iteração 3 à 6

3.4 Seleção da Ferramenta de Gerenciamento de Atividades e Definição do Processo de Desenvolvimento

A equipe de desenvolvimento apresentada anteriormente na seção 3.3, fez uso da ferramenta do **Team Foundation Server** (TFS) [23] para registro de atividades relacionadas a implementação e testes do software.

O TFS é um aglomerado de ferramentas integradas de desenvolvimento de software colaborativas para toda a equipe. Algumas dessas ferramentas atendem a controle de versão, personalização de processos ágeis para gerenciamento e registro de atividade e portal de projeto.

As **sprints** e os **backlogs** do produto são inseridos na ferramenta junto com as atividades referentes a equipe de desenvolvimento para que os mesmos atendam aos requisitos do software, entre outras demandas. Elas variam desde tarefas de implementação de novas funcionalidades, até a correção de **bugs** encontrados na fase de testes. (vide Figura 6).

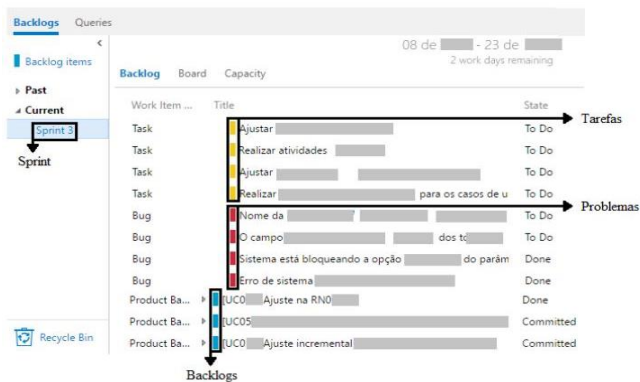


Figura 6. Lista de atividades do TFS. Imagem autorizada da empresa P.

Durante o planejamento da **sprint**, as funcionalidades são analisadas e destrinchadas em tarefas pequenas, onde cada uma delas tem suas regras. Por exemplo, um **backlog** escolhido para implementação, o mesmo demanda a criação de uma tela de cadastro de clientes que salva as informações em um banco de dados através da

internet. As tarefas referentes a essa funcionalidade podem ser divididas pela equipe como: Criação de tela de cadastro de clientes; configurar internet para tráfego de informações de cadastro; e salvar informações do cliente no banco de dados.

Com a funcionalidade destrinchada, cada tarefa é registrada na ferramenta com informações de trabalho estimado e situação inicial de **"A fazer"**, e podem começar a ser implementadas independentemente por qualquer integrante da equipe durante a etapa de execução da **sprint**. Quando isso ocorre, a situação é alterada para **"Em andamento"** e quando a tarefa é finalizada, o trabalho realizado é informado e a situação é alterada para **"Feito"**.

Os **bugs** são problemas encontrados em testes de qualidade do produto (no caso da empresa P, uma equipe de teste é responsável por este processo). Esses problemas podem ser desde um erro no software até a falta de implementação de algum requisito da funcionalidade.

Os problemas são registrados na ferramenta de gerenciamento com a situação inicial de **"A fazer"**. Quando um membro da equipe de desenvolvimento começa a corrigir o **bug**, o esforço estimado é informado e a situação é alterada para **"Em andamento"**. Quando o mesmo é corrigido, o trabalho realizado é informado e a situação é alterada para **"Teste"**. A partir deste ponto o software é testado novamente, onde pode ocorrer do problema persistir, sinalizando que o problema não foi corrigido. Quando isso acontece, o **bug** é reaberto e o fluxo é repetido até a correção ser aprovada pela equipe de teste, alterando assim, a situação para **"Feito"**.

A ferramenta do TFS também guarda o histórico de todas as alterações feitas pelos desenvolvedores e testadores, tornando possível o rastreamento de informações anteriores ao estado atual da atividade. A Figura 7 apresenta uma tela de cadastro e alteração de uma atividade.



Figura 7. Tela de cadastro e alteração de atividade do TFS. Imagem autorizada da empresa P.

3.5 Métodos

O presente trabalho implementa a ludificação em uma equipe de desenvolvimento de software, durante o processo de codificação na empresa P.

Para isso, baseado nas referências da seção 2.5, foi adotado um **framework** de trabalho baseado em desafios, punições e **feedback** para potencializar o trabalho da equipe.

A metodologia se aplica inserindo os elementos de jogos, destacados na seção 2.5, nas atividades da equipe de desenvolvimento que foram registradas na ferramenta de gerenciamento de atividades, descrita na seção 3.4.

Os dados da ferramenta foram coletados e analisados a posteriori e apenas tarefas e problemas com a situação “**Feito**” foram considerados para análise.

Foi também criado neste trabalho um módulo para a mesma ferramenta, denominado **TfsGamified [24]**, que analisa os dados e apresenta ao desenvolvedor informações sobre os resultados e o seu progresso. Trata-se de uma aplicação web que consome os dados do próprio banco de dados do TFS e aplica a ludificação de acordo com a formatação dos elementos de jogos descritos na seção 4. O módulo consulta os registros referente as atividades do projeto de software, bem como os seus registros históricos, e agrupa as informações por desenvolvedor.

Dessa maneira, foi possível identificar quais as atividades concluídas pelo desenvolvedor, e no histórico, se as atividades tiveram sua situação alterada anteriormente e por quem.

4 Experimento

Esta seção apresenta não só os elementos de jogos selecionados para a empresa P, mas também como estes elementos foram formatados para o contexto.

4.1 Sistema de pontuação

O jogador recebe 10 pontos para cada tarefa implementada e problema resolvido.

Caso o jogador tenha concluído alguma dessas atividade e ela tenha sido reaberta, indicando que a tarefa não foi implementada completamente ou que o problema não foi resolvido corretamente:

- Se o mesmo jogador concluir a atividade novamente, em vez de 10, o seu valor será de apenas 5 pontos, como forma de punição.
- Se outro jogador concluir a atividade novamente, ele é quem vai receber a pontuação total (10 pontos), pois foi quem, de fato, a concluiu. O jogador que tivera concluído a atividade anteriormente perde o valor total e recebe apenas ponto.

4.2 Emblemas

São disponibilizados 5 emblemas (vide Figura 8) para os jogadores que se destacam em determinadas situações. Cada um desses emblemas vale 100 pontos e são conquistados de acordo com seus critérios abaixo:

1. O suporte: Concedido ao jogador que obtiver a maior contagem de problemas resolvidos por ele no jogo.
2. O halterofilista: Concedido ao jogador que obtiver a maior contagem de tarefas concluídas por ele no jogo.

3. O certo: Concedido ao jogador que obtiver a maior contagem de atividades feitas dentro do tempo estimado ou com menor diferença (Dif) entre a soma do tempo estimado (Ste) e a do tempo realizado (Str).

$$\text{Dif} = | \text{Ste} - \text{Str} |$$

4. O ajudante: Concedido ao jogador que obtiver a maior contagem de participação em atividades concluídas de outros jogadores.
5. O resolvidor: Concedido ao jogador que obtiver a maior quantidade de atividades concluídas por ele, mas que foram feitas por outros jogadores anteriormente.



Figura 8. Emblemas do jogo. A imagem acima ilustra os emblemas na ordem de apresentação. Imagem elaborada pelo autor.

4.3 Prêmios

São disponibilizados 2 tipos prêmios (vide Figura 9) com o valor de 100 pontos cada, aos jogadores que atendam aos seguintes objetivos:

1. Prêmio de tarefas: Concedido aos jogadores que implementarem 100 ou mais atividades.
2. Prêmio de problemas: Concedido aos jogadores que resolverem 100 ou mais problemas.



Figura 9. Prêmios do jogo. A imagem acima ilustra os prêmios de tarefa e de problema. Fonte: Imagem elaborada pelo autor.

4.4 Classificação

Um ranque de classificação envolvendo todos os jogadores ordenados pela soma (Ptotal) da pontuação adquirida com atividades (Pativ), mais a pontuação conquistada com emblemas (Pemb) e prêmios (Pprem).

$$\text{Ptotal} = \text{Pativ} + \text{Pemb} + \text{Pprem}.$$

4.5 Reputação

O nome dos ganhadores dos emblemas ficam visíveis a todos para que haja reconhecimento entre os jogadores.

4.6 Feedback Imediato

Cada jogador tem acesso a um módulo criado com informações sobre a quantidade de pontos adquiridos, e emblemas e prêmios conquistados. É também apresentada ao jogador, a sua classificação atual, que é de conhecimento privado, para que não haja exposição ou constrangimento.

4.7 Ciclo de Engajamento

É calculado a quantidade de atividades reabertas do jogador, bem como as atividades que os outros jogadores concluíram e que haviam sido concluídas anteriormente por ele. Através dessas informações, é possível analisar o desempenho do jogador e orientá-lo, por meio de notificações, com o intuito de fazê-lo atingir um melhor resultado. As causas e as notificações que podem ser apresentadas são:

- Caso o jogador tenha atividades reabertas: "Você teve várias atividades reabertas e isso faz com que você não alcance a pontuação máxima. Faça mais testes para ter certeza que as concluiu antes de fechá-las".

Ludificação em Engenharia de Software: Tornando o processo de desenvolvimento em uma empresa um jogo

- Caso outros jogadores tenham concluído alguma atividade que, anteriormente, havia sido concluída por ele:
"Após concluir algumas atividades, as mesmas foram reabertas e outro jogador as fechou. Preste mais atenção para não perder os pontos".

5 Resultados

O módulo criado para a ferramenta analisa os dados e apresenta os resultados conforme foram especificados na seção 4. Os dados coletados são de um projeto de software já concluído, por esse motivo, as informações apresentadas a seguir são os resultados referentes a análise final.

5.1 Sistema de Pontuação e Classificação

De acordo com as regras de classificação e pontuação, a primeira posição ficou com Desenvolvedor OE, com 6435 pontos, a segunda com o Desenvolvedor RE, com 5657 pontos, e a terceira com o Desenvolvedor IA, com 4276 pontos. A pontuação e classificação dos demais jogadores podem ser visualizadas na Tabela 3.

Tabela 3. Pontuação e classificação final.

Posição	Desenvolvedor - N°	Pontuação
1°	Desenvolvedor OE - 8	6435
2°	Desenvolvedor RE - 5	5657
3°	Desenvolvedor IA - 13	4276
4°	Desenvolvedor HU - 12	3560
5°	Desenvolvedor LE - 2	3248
6°	Desenvolvedor LA - 1	3055
7°	Desenvolvedor EU - 10	2835
8°	Desenvolvedor UO - 3	1825
9°	Desenvolvedor AA - 7	1024
10°	Desenvolvedor EA - 9	979
11°	Desenvolvedor RU - 4	846
12°	Desenvolvedor HO - 11	465

13°	Desenvolvedor EN - 6	284
-----	----------------------	-----

5.2 Emblemas e Reputação

De acordo com os critérios definidos, os emblemas foram concedidos aos desenvolvedores conforme Tabela 4.

Tabela 4. Quadro de emblemas conquistados.

Emblema	Desenvolvedor - N°
O suporte	Desenvolvedor OE - 8
O halterofilista	Desenvolvedor LA - 1
O certeiro	Desenvolvedor RU - 4
O ajudante	Desenvolvedor OE - 8
O resolvedor	Desenvolvedor RE - 5

Os emblemas de suporte e ajudante foram concedidos ao Desenvolvedor OE, indicando que o mesmo resolveu mais problemas e participou de mais atividades do que os outros jogadores.

O emblema de resolvedor foi concedido ao Desenvolvedor RE, indicando que o mesmo concluiu mais atividades as quais outros jogadores teriam sinalizado como feitas e foram reabertas.

O emblema de halterofilista foi concedido ao Desenvolvedor LA, indicando que o mesmo implementou mais tarefas do que outros jogadores.

O emblema de certeiro foi concedido ao Desenvolvedor RU, indicando que o mesmo possui mais atividades concluídas que foram melhores estimadas.

O módulo apresenta este resultado ao jogador, listando os emblemas e apontando seus respectivos ganhadores (vide Figura 12).

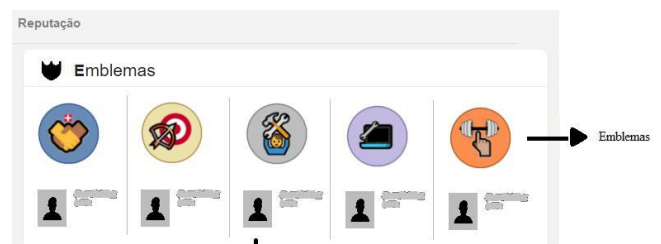


Figura 12. Reputação. A imagem acima apresenta ao jogador informações sobre os ganhadores dos emblemas. Fonte Imagem retirada do módulo criado pelo autor.

5.3 Prêmios

De acordo com os critérios definidos, os prêmios de tarefa e problema foram concedidos aos desenvolvedores conforme Tabela 5.

Tabela 5. Quadro de prêmios conquistados.

Desenvolvedor - N°	Prêmio de tarefas	Prêmio Problema
Desenvolvedor OE - 8	X	X
Desenvolvedor RE - 5	-	X
Desenvolvedor IA - 13	X	X
Desenvolvedor HU - 12	X	X
Desenvolvedor LE - 2	-	X
Desenvolvedor LA - 1	X	X
Desenvolvedor EU - 10	X	X
Desenvolvedor UO - 3	-	-
Desenvolvedor AA - 7	-	-
Desenvolvedor EA - 9	-	-
Desenvolvedor RU - 4	-	-
Desenvolvedor HO - 11	-	-
Desenvolvedor EN - 6	-	-

Como se pode observar, nem todos os jogadores conseguiram completar os desafios impostos de concluir cem ou mais tarefas e problemas. Chama a atenção o caso do Desenvolvedor RE, que não atingiu o objetivo de completar cem ou mais tarefas e ainda assim conquistou uma pontuação que lhe concedeu a segunda colocação. O que pode significar que a

maior parte de sua pontuação veio da resolução de problemas.

5.4 Feedback Imediato e Ciclo de Engajamento

Para atender a esses requisitos foram criadas duas sessões para o módulo da ferramenta de gerenciamento de atividades: Progresso e Estatísticas. Ambas as sessões apresentam ao desenvolvedor informações sobre os seus resultados e desempenho no jogo.

A sessão de Progresso disponibiliza informações sobre a classificação do jogador, a quantidade de pontos, as mensagens de notificação, os emblemas obtidos e prêmios conquistados. Como exemplo, a Figura 10 apresenta o resultado do primeiro colocado, o Desenvolvedor OE.



Figura 10. Progresso. A imagem acima apresenta ao Desenvolvedor OE, o primeiro colocado, informações sobre o seu progresso no jogo. Fonte: Imagem retirada do módulo criado pelo autor.

Também é apresentado na sessão de Progresso um painel de notificações, que informa ao desenvolvedor se o mesmo possui pontos perdidos por conta de atividades reabertas ou concluídas por outros desenvolvedores. Essas informações são quantificadas e apresentadas ao mesmo na sessão de Estatísticas.

Esses números representam 16% e 10% respectivamente, do total de atividades concluídas pelo desenvolvedor.

6 Conclusão e Trabalhos Futuros

A ludificação não é uma técnica fácil de ser aplicada em equipes de desenvolvimento no ambiente corporativo.

Foi necessário identificar quais atividades levam a atingir o sucesso na construção de um software de qualidade, e assim, definir quais elementos de jogos são mais adequados para tentar engajar e melhorar o desempenho entre os integrantes.

Os elementos de jogos foram formatados para se adaptarem ao fluxo de trabalho desempenhado pela equipe de desenvolvimento, com o intuito de fazê-la atingir melhores resultados, orientando nas melhores práticas em relação as atividades cadastradas na ferramenta de gerenciamento. O sistema de pontuação, por exemplo, pode fazer com que o desenvolvedor busque concluir mais atividades com mais qualidade, minimizando os riscos delas serem reabertas. Os prêmios podem servir como uma medalha dada quando os objetivos traçados forem alcançados, motivando o desenvolvedor. E a reputação adquirida através de emblemas pode levar o desenvolvedor a se tornar referência em determinadas situações. Tais informações podem ser de extrema importância, tanto para a equipe, quanto para líderes e gerentes de projeto, pois os resultados e a quantificação das atividades representam o desempenho individual de cada desenvolvedor em relação ao universo do projeto de software.

Devido aos artefatos gerados e a organização do seu fluxo de trabalho, a adoção do Scrum como metodologia ágil pela equipe de desenvolvimento fez com que existisse uma diretriz para a rastreabilidade e observação das suas atividades. Desse modo, cria-se um cenário ideal para a aplicação ludificação de forma automática, sem causar impacto nos exercícios das funções já praticadas pela equipe. Isso faz com que a ludificação seja aplicada de forma imersiva.

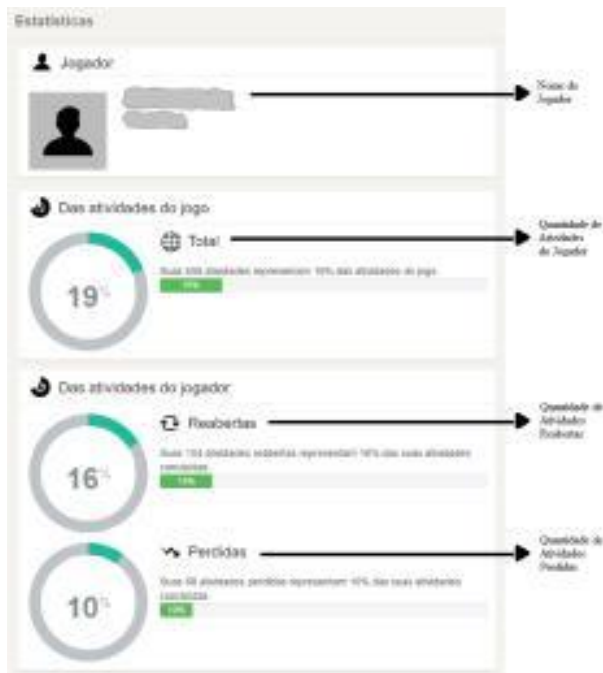


Figura 11. Estatísticas. A imagem acima apresenta ao Desenvolvedor OE, o primeiro colocado, informações quantificadas sobre as suas atividades no projeto de software. Fonte: Imagem retirada do módulo criado pelo autor.

A sessão de Estatísticas apresenta ao desenvolvedor a quantidade de atividades concluídas por ele e o percentual que esse valor representa em relação a quantidade total de atividades analisadas do projeto de software. Nessa mesma sessão, é apresentada também a quantidade de atividades reabertas e concluídas por outros desenvolvedores e o percentual que esses valores representam em relação a quantidade de atividades concluídas por ele. Como exemplo, a Figura 11 demonstra que as 658 atividades concluídas pelo primeiro colocado, o Desenvolvedor RE, representam 19% das atividades totais do jogo. Dessas atividades concluídas, 104 foram reabertas e 68 foram concluídas por outros jogadores, mas que anteriormente teriam sido concluídas por ele, significando que o desenvolvedor deixou de ganhar a pontuação máxima nessas atividades.

A criação do módulo para a ferramenta de gerenciamento de atividades, no intuito de auxiliar na visualização dos resultados, é uma abordagem que faz com que a participação dos desenvolvedores seja natural. Como a aplicação apenas consome os dados da ferramenta e os analisa, tais informações são apresentadas de forma dinâmica e poderiam ser visualizadas por qualquer membro da equipe, em qualquer período dentro do fluxo de implementação do projeto de software. Dessa maneira, não é preciso finalizá-lo para se ter acesso aos resultados. Contudo, esse cenário não representaria o resultado final, pois poderia sofrer alteração à medida em que a equipe conclui suas atividades.

Como trabalhos futuros, é proposto a aplicação e acompanhamento do módulo criado, que implementa a ludificação no TFS, em um projeto de software que ainda está em andamento em uma empresa para avaliar a contribuição do mesmo na Engenharia de Software. Também é proposto uma extensão do módulo de forma que os líderes de equipe e gerentes de projeto tenham acesso à ela para poderem acompanhar o andamento das atividades e o progresso da equipe.

Referências

- [1] J. McGonigal. Reality is Broken - Why Games Make Us Better and How They Can Change the World. The Penguin Press, New York, 2011.
- [2] T. Malone. What Makes Things Fun to Learn? A Study of Intrinsically Motivating. Computer Games, Palo Alto, 1980.
- [3] S. Deterding, . Gamification: Design for motivation. **Interactions**, 19:14-17, 2012.
- [4] R. Smith. Game Impact Theory: The Five Forces That Are Driving the Adoption of Game Technologies within Multiple Established Industries. U.S. Army Program Executive Office for Simulation, Training, and Instrumentation, Orlando, 2008.
- [5] E. Mollick, N. Rothbard. Mandatory Fun: Consent, Gamification and the Impact of Games at Work. University of Pennsylvania, Set. 2014.
- [6] A. Marczewski. Gamification: A Simple Introduction 2nd ed. Amazon, 2013.
- [7] E. Passos, D. Medeiros. Turning RealWorld Software Development into a Game. SBC Proceedings of SBGames, Salvador, 2011.
- [8] I. Marsic. Software Engineering. Rutgers, New Jersey, 2012.
- [9] I. Sommerville. Software Engineering 9th ed. Pe-arson, Boston, 2011.
- [10] IBM. Rational Unified Process - Best Practices for Software Development Teams. Rational Software White Paper, 2001.
- [11] V. Szalvay. An Introduction to Agile Software Development. Danube Technologies Inc, Bellevue, 2004.
- [12] K. Schwaber. Guia do Scrum. Scrum Alliance, 2009.
- [13] S. Kenneth. Essential Scrum - A Practical Guide To The Most Popular Agile Process. Pearson, Michigan , 2012.
- [14] S. Deterding, D. Dixon. From Game Design Elements to Gamefulness: Defining "Gamification". Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, 2011.
- [15] A. Yohannis, D. Prabwod., A. Waworuntu. Defining Gamification: From lexical meaning and process viewpoint towards a gameful reality. International Conference on Information Technology Systems and Innovation, Bangdung, 2014.
- [16] G. Zichermann, C. Cunningham. Gamification by Design – Implementing Game Mechanics in Web and Mobile Apps. O’Reilly, Sebastopol , 2011.
- [17] E. Kanat, S. Siloju, T. Raghu, S. Vinze. Gamification of emergency response training: A public health example. International Conference on Intelligence and Security Informatics, Seattle, 2013.
- [18] M. Kapp. The Gamification of Learning And Instruction: game-based methods and strategies for training and education. Pfeiffer, San Francisco, 2012.
- [19] O. Pedreira, F. Gacia, N. Brisaboa, M. Piattini. Gamification in software engineering - A

systematic mapping. Information and Software Technology, 57:157-168, 2015

[20] R. Cavalcanti. Ludificação na Engenharia de Software: Aplicação no Agilo For Trac. Universidade de Pernambuco, 2016.

[21] 2017Agilo For Trac. Disponível em <https://www.agi-lofortrac.com>. Acesso em: 18 jul. 2017.

[22] Porto Digital. Disponível em: <http://www.porto-digital.org>. Acesso em: 14 jun. 2017.

[23] Team Foundation Server. Disponível em <https://www.visualstudio.com/pt-br/tfs/>. Acesso em: 19 abr. 2017.

[24] TfsGamified. Disponível em <https://github.co-m/joelnetodev/tfsgamified>. Acesso em: 19 abr. 2017.