

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Game Design Techniques for Software Engineering

Rafaela Marques Pinto de Faria



Mestrado Integrado em Engenharia Informática e Computação

Orientador: Ana Cristina Ramada Paiva

Junho de 2016

© Rafaela Faria, 2016

Game Design Techniques for Software Engineering

Rafaela Marques Pinto de Faria

Mestrado Integrado em Engenharia Informática e Computação

27 de junho de 2016

Resumo

A Engenharia de Software é uma área da computação que aplica a teoria, o conhecimento e a prática, de forma articulada e contextualizada, no desenvolvimento de sistemas de software que satisfaçam os requisitos do usuário de forma eficiente. Tem uma vasta densidade de conteúdos sendo, por isso, organizada numa base de conhecimento pelo IEEE (Instituto de Engenheiros Eletrotécnicos e Eletrônicos), de forma a sistematizar os conceitos neles presentes, tornando-os mais acessíveis e operacionalizáveis. A área da Engenharia de Software constitui-se como um desafio no ensino devido, por um lado, à sua amplitude de conceitos e subáreas, que potencia a sua abrangência, e, por outro, proporciona uma abordagem muito singular de cada tópico, enfatizando a sua vertente de especificidade.

O exercício com jogos é cada vez mais procurado pelos agentes responsáveis do ensino. O uso desta ferramenta educativa incentiva a aprendizagem através de atividades lúdicas e atrativas, tornando a captura de conhecimento mais eficaz, além de, em virtude do seu caráter mais prático, contribuir também para uma consolidação mais facilitada. Para o desenvolvimento desses jogos ser objetivo e coerente, é necessário identificar uma série de padrões e técnicas de desenho, de forma a cobrir os objetivos pedagógicos da área a ser ensinada.

Além disso, o mapeamento de padrões de desenho de jogos para os objetivos pedagógicos da Gestão de Engenharia de Software proporciona condições para que o desenvolvimento de jogos use e otimize técnicas que conseguem cobrir os tópicos de ensino da área. A criação de um mapa semântico para a subárea dos Requisitos de Software, do mesmo modo, vai permitir que os novos jogos desta área também cumpram todos os objetivos pedagógicos a que se propõe.

Nesta dissertação pretende-se analisar os jogos de Engenharia de Software existentes, de forma a perceber onde são usados os padrões de desenho de jogos. O desenvolvimento de um mapa de padrões de desenho de jogos para os tópicos dos Requisitos de Software e a proposta de melhorias para jogos encerram os objetivos deste trabalho de investigação.

Abstract

Software Engineering is the field in computation which applies the theory, knowledge and practice in the development of software systems that efficiently meet user requirements. It has a wide variety of contents. Therefore, in order to systematize ideas, IEEE compiled a knowledge base with the existing concepts. The field of Software Engineering is a challenging topic when it comes to teaching. Its variety of concepts and large number of subareas entails a very unique approach to each topic.

The use of games is increasingly popular in education. This tool encourages learning through play and attractive activities, making knowledge more accessible. For the development of these games to be objective and consistent, it is necessary to identify a set of patterns and design techniques, in order to cover the pedagogical goals of the area to be taught.

The mapping of game design patterns for the pedagogical goals of Software Engineering Management allows the development of games to be based on techniques that can cover the teaching topics in the area. The creation of a semantic map for the subarea of Software Requirements will allow the future games of this area to also meet the proposed pedagogical objectives.

This thesis aims to analyze existing Software Engineering games, in order to understand where game design patterns are used. The development of a map of game design patterns for the topics of Software Requirements, and the proposed improvements for games conclude the objectives of this research work.

Agradecimentos

Esta dissertação é a conclusão de uma caminhada de vários anos, onde inúmeras pessoas me ajudaram a atingir metas, das mais variadas formas.

Agradeço à Faculdade de Engenharia da Universidade do Porto e a todos os professores, e outros profissionais, que me apoiaram ao longo destes anos.

Obrigada professora Ana Paiva, pela paciência e dor de cabeça que teve comigo durante o desenvolvimento desta dissertação. Obrigada pela resposta aos pedidos de ajuda e pela motivação que fez questão de me passar. Todo o seu apoio foi essencial para a conclusão deste trabalho.

Ao professor Nuno Flores, agradeço todas as opiniões e colaboração no desenrolar de atividades inerentes a esta investigação.

Aos profissionais de Engenharia de Requisitos e ao alunos de Laboratório de Programação Orientada por Objetos, muito obrigada pela participação na experiência que realizei.

Agradeço do fundo do coração aos meus pais, as duas pessoas sempre presentes e incansáveis durante toda a minha vida. Obrigada por todos os valores que me transmitiram e todo o amor que depositaram na minha educação. O vosso apoio é fundamental em todas as etapas da minha vida.

Henrique, nunca vou conseguir agradecer todo o apoio que me deste. Obrigada por nunca desistires de me empurrar para a frente e por teres sido um ponto fundamental na conclusão deste curso. O meu eterno obrigada.

Toda a minha família merece uma palavra de apreço pelo amor incondicional que me dão. Somos a melhor família do mundo e são todos muito importantes para mim.

À minha Jessica um astronómico obrigada. Agradeço-te todo o conforto que me dás quando preciso e toda a força que depositas na nossa amizade. Obrigada por me receberes sempre de braços abertos e por seres a melhor amiga que todos querem e precisam de ter. O teu apoio na escrita desta dissertação foi fundamental.

Ao Pedro Faria agradeço a amizade durante estes anos, que foi muito importante para mim. As aventuras que passámos juntos e os nossos diálogos vão comigo para a vida.

Obrigada Nelson por tudo e mais alguma coisa. És uma pessoa extraordinária e uma mais valia para o mundo. A tua amizade é essencial e todas as opiniões que partilhamos fazem-me crescer sempre mais um bocadinho.

Agradeço a todos os meus colegas, tanto de Engenharia Eletrotécnica como de Engenharia Informática, que foram sempre impecáveis e amigos. Edu, Esteves, Carlos, Rafa, Diana, Pipo, Ratatui e Mário.

Agradeço ao mano João por tudo. Obrigada por me teres ajudado a crescer e por me teres passado um bocadinho da pessoa que és. Sempre foste insubstituível na minha vida. Agradeço o esforço depositado na correção desta tese e todo o apoio ao longo desta etapa.

Obrigada Ana Maria por seres uma amiga extraordinária. Agradeço-te todos os momentos que temos vindo a partilhar e todos os que ainda se encontram pela frente. Obrigada por estares sempre pronta a ajudar-me em tudo o que preciso.

Agradeço às Alheiras pela amizade verdadeira e por me fazerem esquecer todas as coisas menos boas quando estamos juntos. O convívio convosco é uma lufada de ar fresco nesta vida que teima em chatear-nos a todos. São os melhores amigos que alguém pode ter.

“It is not the strongest of the species that survive,
nor the most intelligent, but the ones most responsive to change.”

Charles Darwin

Conteúdo

| | |
|--|-----------|
| Introdução..... | 1 |
| 1.1 Contexto e Problema..... | 1 |
| 1.2 Movivação..... | 2 |
| 1.3 Objetivos..... | 2 |
| 1.4 Estrutura do Documento..... | 3 |
| Estado da Arte..... | 5 |
| 2.1 Ensino da Engenharia de Software..... | 5 |
| 2.2 Jogos Sérios no Ensino..... | 8 |
| 2.3 Padrões Pedagógicos..... | 11 |
| 2.4 Padrões de Desenho para Jogos..... | 13 |
| 2.5 Mapeamento de Padrões de Desenho de Jogos..... | 17 |
| 2.6 Resumo e Conclusões..... | 20 |
| Papel da Engenharia de Software no Ensino..... | 23 |
| 3.1 Análise Comparativa de Jogos Sérios de Engenharia de Software..... | 23 |
| 3.2 Mapeamento de Padrões para o Ensino de Requisitos de Software..... | 33 |
| 3.3 Proposta de Melhorias a Jogos Sérios de Engenharia de Software..... | 42 |
| 3.4 Resumo e Conclusões..... | 44 |
| Conclusões e Trabalho Futuro..... | 47 |
| 4.1 Satisfação dos Objetivos..... | 48 |
| 4.2 Trabalho Futuro..... | 48 |
| Referências..... | 51 |
| A Inquérito a Profissionais de Requisitos de Software..... | 55 |
| B Médias dos Participantes na Experiência..... | 79 |
| C Pré-Questionário da Experiência..... | 81 |
| D Questionário de Fatores Externos e Satisfação Geral da Experiência..... | 85 |
| E Questionário sobre Conhecimentos em Engenharia de Software..... | 91 |

Lista de Figuras

| | |
|---|----|
| Figura 1 - Pré requisitos para uma aprendizagem eficaz [PPP15]..... | 11 |
| Figura 2 - Abordagem ao mapeamento [Let15]..... | 18 |
| Figura 3 - Resultados dos questionários da experiência [Let15]..... | 20 |
| Figura 4 - Delineação da experiência..... | 37 |
| Figura 5 - Resultados do questionário do grupo A (não jogou o jogo) e do grupo B (jogou o jogo antes de responder)..... | 41 |

Lista de Tabelas

| | |
|--|----|
| Tabela 1 - Mapeamento das funções de aprendizagem de "Preparação" [KKS11] | 14 |
| Tabela 2 - Mapeamento das funções de aprendizagem de "Manipulação de conhecimento" [KKS11] | 14 |
| Tabela 3 - Mapeamento das funções de aprendizagem de "Relações de ordem superior" [KKS11] | 15 |
| Tabela 4 - Mapeamento das funções de aprendizagem de "Regulação do aluno" [KKS11]..... | 15 |
| Tabela 5 - Mapeamento das funções de aprendizagem de "Ações produtivas" [KKS11]... | 16 |
| Tabela 6 - Padrões de desenho de conteúdos [MLM11] | 17 |
| Tabela 7 - Mapeamento dos objetivos pedagógicos de Gestão de Engenharia de Software para padrões de desenho de jogos [Let15] | 18 |
| Tabela 8 - Análise de Jogos Sérios de Engenharia de Software | 24 |
| Tabela 9 - Padrões para elementos de jogo nos Jogos Sérios selecionados | 25 |
| Tabela 10 - Padrões para recursos e gestão nos Jogos Sérios selecionados | 26 |
| Tabela 11 - Padrões para informação, comunicação e apresentação nos Jogos Sérios selecionados | 26 |
| Tabela 12 - Padrões para ações e eventos nos Jogos Sérios selecionados | 27 |
| Tabela 13 - Padrões para estruturas narrativas, previsão e imersão nos Jogos Sérios selecionados | 27 |
| Tabela 14 - Padrões para interação social nos Jogos Sérios selecionados | 28 |
| Tabela 15 - Padrões para objetivos nos Jogos Sérios selecionados | 28 |
| Tabela 16 - Padrões para estruturas objetivo nos Jogos Sérios selecionados | 29 |
| Tabela 17 - Padrões para sessões de jogo nos Jogos Sérios selecionados | 29 |

| | |
|--|----|
| Tabela 18 - Padrões para mestria do jogo e equilíbrio nos Jogos Sérios selecionados | 30 |
| Tabela 19 - Padrões para repetibilidade de meta jogos e curvas de aprendizagem nos Jogos Sérios selecionados | 30 |
| Tabela 20 - Funções de aprendizagem de preparação usadas nos Jogos Sérios selecionados | 31 |
| Tabela 21 - Funções de aprendizagem de manipulação de conhecimento usadas nos Jogos Sérios selecionados | 31 |
| Tabela 22 - Funções de aprendizagem de relações de ordem superior usadas nos Jogos Sérios selecionados | 31 |
| Tabela 23 - Funções de aprendizagem de regulação do aluno usadas nos Jogos Sérios selecionados | 32 |
| Tabela 24 - Funções de aprendizagem de ações produtivas usadas nos Jogos Sérios selecionados | 32 |
| Tabela 25 - Resultados obtidos pela análise do inquérito aos especialistas | 34 |
| Tabela 26 - Mapeamento dos tópicos dos Requisitos de Software para as funções de aprendizagem e ensino de Shuell e Moran e padrões de desenho de jogos | 35 |
| Tabela 27 - Dados relativos aos grupos participantes na experiência..... | 36 |
| Tabela 28 - Resultados do pré-questionário | 38 |
| Tabela 29 - Resultados do questionário sobre fatores externos | 39 |
| Tabela 30 - Resultados do questionário de satisfação geral | 40 |
| Tabela 31 - Resultados obtidos pelo grupo A e grupo B no questionário de conhecimento..... | 40 |
| Tabela 32 - Médias dos estudantes que participaram na experiência | 80 |
| Tabela 33 - Resultados do pré-questionário do grupo A | 83 |
| Tabela 34 - Resultados do pré-questionário do grupo B | 84 |
| Tabela 35 - Resultados do questionário de fatores externos do grupo B | 88 |
| Tabela 36 - Resultados do questionário de satisfação geral do grupo B | 89 |
| Tabela 37 - Resultados do questionário de conhecimento do grupo A | 95 |
| Tabela 38 - Resultados do questionário de conhecimento do grupo B | 96 |

Abreviaturas e Símbolos

| | |
|--------|---|
| SWEBOK | Software Engineering Body of Knowledge |
| IEEE | Instituto de Engenheiros Eletrotécnicos e Eletrónicos |
| KA | Áreas de Conhecimento |

Capítulo 1

Introdução

A Engenharia de Software é uma área da computação que tem como objetivo a especificação, desenvolvimento e manutenção de sistemas de software. É uma área com um grande mercado e variedade de conceitos, estando em permanente renovação a forma como é ensinada. Com a densidade de tópicos existente, é preciso escolher diferentes formas de os abordar, para que haja uma aprendizagem efetiva e produtora.

Jogar jogos, nas suas diversas formas, sempre foi uma atividade aliciante. Os jogos educativos vieram, por isso, revolucionar o mundo do ensino. Desde crianças a exercitar o cálculo matemático, a adultos a aprender inglês, entre muitas outras possibilidades, as pessoas têm beneficiado dos chamados Jogos Sérios. Eles são cada vez mais populares, já que permitem o desenvolvimento do intelecto através de atividades de carácter lúdico, onde o fator saturação é substancialmente reduzido.

1.1 Contexto e Problema

Esta dissertação baseia-se num trabalho já realizado e cujo objetivo foi encontrar uma maneira mais interessante de aplicar o uso de Jogos Sérios na aprendizagem de Gestão de Engenharia de Software, uma das suas áreas. As metas propostas foram cumpridas e resultaram num mapeamento semântico dos objetivos educacionais e pedagógicos de Gestão de Engenharia de Software para padrões de desenho de jogos. Este mapeamento serve de orientação para o desenho de Jogos Sérios desta área. A sua importância advém da complexidade na criação deste tipo de jogos, que beneficiam de um desenho claro dos objetivos a cumprir.

Como a solução do trabalho anterior só está aplicada a uma área da Engenharia de Software, e tendo em conta o sucesso obtido, faz sentido validar o mapeamento para as outras áreas.

1.2 Motivação

A Engenharia de Software usa muitas áreas diferentes para além da computação, como a matemática ou a gestão. Como é um domínio tão importante, é necessário abordá-lo de uma forma muito pragmática, sendo fundamental perceber com clareza os conceitos e aplicá-los de forma útil e verdadeiramente ajustada à variabilidade de situações possíveis.

Os Jogos Sérios revestem-se de um enorme potencial para aquisição de conhecimento, tendo, naturalmente, fruto das suas características, uma forte componente motivacional, que desperta a atenção e estimula o jogador a atingir aquele desiderato.

Os padrões e técnicas de desenho de jogos são essenciais para o seu desenvolvimento. As diferentes áreas da Engenharia de Software são distintas e têm objetivos de ensino muito específicos, de acordo com as particularidades de cada uma. Para se desenvolver conteúdos educativos para jogos, é essencial ter estas questões em conta. Assim, em síntese, considera-se que a motivação desta investigação se centra em identificar os padrões e técnicas de design de jogos que melhor encaixam no ensino dos conteúdos da área dos Requisitos de Software, uma das subáreas da Engenharia de Software.

1.3 Objetivos

O objetivo principal deste projeto é dar continuidade a um trabalho de investigação realizado anteriormente. Numa primeira abordagem foi produzido um mapa semântico dos objetivos educacionais e pedagógicos de Gestão de Engenharia de Software para padrões de design de jogos, pretendendo-se estender a metodologia para os Requisitos de Software. Nesse contexto, torna-se determinante reformular os artefatos para os aplicar à nova área. Além disso, é importante validar esta extensão, de forma a ser facilitada a definição de uma metodologia extensível a várias áreas da Engenharia de Software.

A investigação pretende também fazer um levantamento e análise comparativa de Jogos Sérios de aprendizagem de Engenharia de Software, com o objetivo de perceber o mercado dos

mesmos. É relevante entender em que áreas da Engenharia de Software é que se focam, bem como a linguagem em que foram programados, ou se têm a API disponível.

Também faz parte deste trabalho verificar o que falta abordar nos Jogos Sérios disponíveis que ensinam Engenharia de Software. A partir deles, procurar-se-ão os padrões de desenho de jogos existentes e verificar-se-á se estão de acordo com os objetivos educacionais e pedagógicos desejados. A finalidade é propôr melhorias de forma a cobrir esses objetivos, com a inclusão de padrões que vão de encontro ao que falta colmatar.

Como resultado final deste trabalho pretende-se construir um guia que auxilie quem vai desenhar um jogo sério para o ensino de Requisitos de Software, de forma a desenvolver uma ferramenta educativa que vá de encontro aos objetivos pedagógicos dessa área.

1.4 Estrutura do Documento

O documento encontra-se estruturado em quatro capítulos principais.

No presente capítulo temos a introdução do trabalho, apresentando o seu contexto, a motivação e os objetivos.

No segundo capítulo é apresentado o estado da arte. Abordamos a Engenharia de Software e o seu ensino, os Jogos Sérios, os padrões pedagógicos e os padrões de design de jogos educativos.

O terceiro capítulo aborda o mapeamento dos padrões de desenho de jogos para a área dos Requisitos de Software, as tabelas de análise comparativa dos jogos que ensinam Engenharia de Software e as melhorias propostas para os jogos.

Por último, são apresentadas as conclusões e a direção do trabalho, a desenvolver posteriormente.



Capítulo 2

Estado da Arte

2.1 Ensino da Engenharia de Software

A Engenharia de Software é uma disciplina que aplica a teoria, o conhecimento e a prática, no desenvolvimento de sistemas de software para que satisfaçam os requisitos dos usuários de forma eficiente [ACM04].

A SWEBOK [BF14], uma abreviação para *The Guide to the Software Engineering Body of Knowledge*, é um documento criado pelo Instituto de Engenheiros Eletrotécnicos e Eletrônicos (IEEE) com a finalidade de servir de referência a assuntos presentes na Engenharia de Software.

No SWEBOK são identificadas 10 áreas relacionadas com a Engenharia de Software:

- Ciência da Computação;
- Engenharia de Computadores;
- Engenharia de Sistemas;
- Ergonomia de Software;
- Gestão;
- Gestão da Qualidade;
- Gestão de Projeto;
- Matemática;
- Sistemas de Informação.

Estas áreas são usadas, de alguma forma, processos de Engenharia de Software, seja para o desenvolvimento de produtos como para a sua gestão.

Também são apresentadas as 15 áreas de conhecimento (KA) de Engenharia de Software, que são responsáveis por todo o desenvolvimento de um produto de Software:

- Requisitos de Software:
 - Requisitos funcionais, correspondem à funcionalidade do software;
 - Requisitos não funcionais, expressam as restrições e características que o software deve ter;
 - Requisitos inversos, definem estados e situações que nunca podem acontecer;
- Design de Software;
- Construção de Software;
- Teste de Software:
 - Teste funcional, verifica as regras de negócio e condições válidas e não válidas;
 - Teste de recuperação de falhas, com falhas provocadas a fim de verificar eficiência;
 - Teste de desempenho, verifica o tempo de resposta e processamento para diferentes configurações;
 - Teste de segurança e controlo de acesso, verifica a proteção de acesso aos dados;
 - Teste de interfaces, com a navegação, consistência e padrões;
 - Teste de volume, verifica até que ponto o software aguenta sem crashar;
- Manutenção de Software:
 - Manutenção corretiva, modifica o software para correção de erros;
 - Manutenção adaptativa, altera o software para acomodar mudanças em ambiente externo;
 - Manutenção perfetiva, para aprimorar o software, com modificações não previstas nos requisitos iniciais;
 - Manutenção preventiva, modificando o software para o tornar mais fácil de ser corrigido, adaptado e melhorado;
- Gestão de Configuração de Software;
- Gestão de Engenharia de Software;
- Engenharia de Processo de Software;
- Ferramentas e Métodos de Software;
- Qualidade de Software
- Prática Profissional de Engenharia de Software;
- Economia de Engenharia de Software;

- Fundações da Computação;
- Fundações da Matemática;
- Fundações da Engenharia.

Ainda no SWEBOK são detalhados os tópicos em que os Requisitos de Software se especificam.

- Fundamentos dos Requisitos de Software, onde se define requisito de software, os requisitos de produto e de processo, os requisitos funcionais e não funcionais, as propriedades emergentes, os requisitos quantificáveis e de sistema e requisitos de software;
- Processo de Requisitos, onde se encontra o modelo e atores de processo, suporte e gestão de processo e também a qualidade e melhoria de processo;
- Elicitação de Requisitos, com as fontes de requisitos e as técnicas de elicitação;
- Análise de Requisitos, onde se classifica os requisitos, o modelo conceptual, a arquitetura, alocação e negociação de requisitos e a análise formal;
- Especificação de Requisitos, definindo o documento de definição e especificação de requisitos do sistema e a especificação de requisitos de software;
- Validação de Requisitos, onde se faz a revisão dos requisitos, o protótipo, o modelo de validação e os testes de aceitação;
- Considerações Práticas, definindo a natureza iterativa do processo de requisitos, a gestão da mudança e os atributos, o desenho e a medição de requisitos;
- Ferramentas de Requisitos de Software, onde existem ferramentas de modelagem e ferramentas para a gestão de requisitos.

O ensino da Engenharia de Software passa por aplicar métodos educativos apropriados a cada área. No caso dos Requisitos de Software, Macaulay e Mylopoulos [MM95] argumentam um dilema educacional. Para uma base sólida de conhecimento nesta matéria, é necessária a exposição a incertezas inerentes e inconsistências de problemas reais. Torna-se uma aprendizagem desafiante por ser necessário ir ao foco da realidade.

A forma como se ensina Engenharia de Software, nem sempre vai de encontro ao que um engenheiro de Software deve saber. Esse é um grande problema no ensino desta área, e advém do facto de se expôr maioritariamente conceitos teóricos aos estudantes e dos limites práticos de um projeto na sala de aula [Oh02]. Oh referiu ainda que o ensino da Engenharia de Software tem que ultrapassar o uso de projetos na sala de aula e apoiar-se no uso de simulações de processos de Software.

A área da Engenharia de Software apresenta desafio no ensino devido à sua amplitude de conceitos e sub áreas. Com a sua variedade de tópicos, é preciso perceber qual a melhor forma de ensinar cada um. Como são muito distinto, têm que ser abordados de maneiras diferentes.

Como este trabalho de investigação se foca no ensino da Engenharia de Software, foi relevante perceber em que é que esta área se baseia, bem como onde e a forma como é usada.

2.2 Jogos Sérios no Ensino

Os jogos digitais são uma forma de divertimento. Segundo Prensky [Pre01] há uma série de razões que dão muito potencial aos jogos digitais. Para além de serem um jogo, despertando um envolvimento mais intenso, têm uma estrutura bem definida, induzida pelas regras inerentes a esta atividade. Estes jogos são objetivos, interativos e adaptativos, transmitindo e despertando sensações no jogador. Os desafios ou a competição, aliciam o sujeito a melhorar o seu desempenho. Os estados de vitória despertam gratificação, o que garante um contínuo estado de motivação. Estes jogos têm ainda uma forte componente social, para envolver o jogador com outras pessoas que partilhem os mesmos desafios. Os jogos de tabuleiro, filmes e livros também englobam alguns destes atributos, no entanto, os jogos digitais albergam todas essas características em simultâneo, o que os torna, se bem usados e adaptados, uma ferramenta educativa de grande qualidade.

Como os jogos digitais são tão cativantes, achou-se interessante aproveitá-los no âmbito dos Jogos Sérios ou educativos, onde se aprende alguma temática à medida que se joga. O objetivo deste tipo de jogos é passar conhecimento. Falando da sua influência positiva, Mitchell e Savill-Smith [MS04] mostram as competências que este tipo de atividades ajuda a desenvolver:

- Capacidade analítica e espacial;
- Capacidade estratégica e de reconhecimento;
- Capacidade de captação de conhecimento e aprendizagem;
- Capacidade psicomotora;
- Seletividade visual;
- Atenção.

Também podem influenciar positivamente nos seguintes aspetos:

- Auto monitorização;
- Capacidade de reconhecimento;
- Capacidade de resolução de problemas;
- Tomada de decisão;
- Memória de curto e longo prazo;
- Capacidades sociais como colaboração, negociação e tomada de decisões em grupo.

A primeira definição de Jogo SériO foi introduzida por Abt [Abt70], que os descreveu como simulações e jogos para complementar a educação tanto dentro como fora do contexto da sala de aula. Mais tarde, a definição foi redefinida. Para Sawyer [Saw02] o conceito passava pela ideia de usar o objetivo sério do conhecimento e aprendizagem, aproveitando tecnologias e interesses da indústria dos jogos de computador. Atualmente, as definições de Jogo SériO baseiam-se na ideia de Sawyer, mas os limites do domínio onde ele se encontra são muito subjetivos. Os Jogos Sérios estão presentes, tanto na educação, como na medicina, política ou mesmo na publicidade. Contudo, estas diferentes áreas nem sempre concordam com o que deve ou não estar presente num jogo sério, como diz Corti [Cor06].

Segundo Djaouti, Alvarez e Jessel [DAJ11] os Jogos Sérios são classificados segundo o modelo G/P/S. Esta classificação baseia-se em três aspetos:

- Jogabilidade “gameplay”, que engloba a estrutura do jogo e como é jogado. É diferenciada pelo tipo, objetivos e recursos pretendidos;
- Propósito “purpose”, para avaliar os objetivos que não são de entretenimento;
- Âmbito “scope”, definindo para quem se dirige o jogo.

Com estes três conceitos, os jogos podem ser desenhados para se adequarem ao que o utilizador procura.

Djaouti, Alvarez e Jessel, diferenciaram os Jogos Sérios dos de entretenimento pelo propósito sério que eles apresentam. Segundo os autores, um jogo de entretenimento só apresenta a vertente do jogo, não tendo um propósito sério ou educacional explícito.

Já Susi [SJB07] diferencia estes dois tipos de jogos consoante 4 tópicos nos quais eles se baseiam:

- Tarefa principal do jogo;
- Foco;
- Simulação das atividades presentes;
- Comunicação entre o jogo e o jogador.

O objetivo dos Jogos Sérios é resolver problemas propostos. Focam-se em abordar elementos importantes da aprendizagem e a aplicar pressupostos corretos na simulação de atividades. Como o jogo tem uma vertente muito importante de transmissão de conhecimento, a comunicação com o jogador deve ser o mais realista possível, para não introduzir erros na aprendizagem. Os jogos de entretenimento optam por experiências de jogo mais ricas e foco na recreação do jogador. A simulação de atividades segue processos simplificados e a comunicação com o jogador é perfeita, não se aproximando da realidade. O objetivo desta comunicação perfeita é envolver o jogador dentro do jogo, em vez de lhe passar informação do ecrã para a realidade.

Há vários exemplos de Jogos Sérios, desde jogos de computação, como o *SimSE*, a jogos de medicina, como o *Pulse!!!*.

Este trabalho de investigação foca-se na investigação de jogos sérios no ensino da Engenharia de Software e, para além de se perceber o que são jogos sérios e como mudam a forma como alguém adquire conhecimento, também foi importante analisar o mercado de jogos existentes na área escolhida.

Foram analisados vários Jogos Sérios para ensino de Engenharia de Software:

- *SimSE* [Nav06] é um jogo de simulação para ensinar Gestão de Engenharia de Software. O jogador é um gestor de projetos e o seu objetivo é gerir a sua equipa para desenvolver um produto de software, despedindo elementos da sua equipa ou aumentando o ordenado de um empregado. É suposto seguir um processo de Software definido, como o caso do modelo em cascata;
- *SE RPG* [BM08] usa a mesma abordagem do SimSE de Gestão de projetos. Este jogo permite aos estudantes experimentar o papel de um gestor num processo de Software, seguindo um modelo de processo de Software;
- *PlayScrum* [FS10] é um jogo de cartas para vários jogadores, onde cada um deles é o *Scrum Master* do desenvolvimento de um produto de Software, seguindo as práticas de *Scrum*. O primeiro jogador a completar todas as tarefas sem erros, ou o jogador com maior percentagem de tarefas sem erros, no fim da última iteração, é o vencedor;
- *U-Test* [Sil10] aborda o ensino de teste de Software através de um jogo de simulação. O jogador é responsável por aplicar o teste de unidade a funções que se encontram no jogo;
- *SESAM* [DJ00] é um jogo de Gestão de Engenharia de Software, onde o estudante tem que gerir um processo de desenvolvimento de software. É da responsabilidade do jogador atingir determinados níveis de qualidade nos vários artefactos do jogo;
- *AMEISE* [Mit03] usa a mesma abordagem do SESAM mas acrescenta a componente de multijogador;
- *Sim JavaSP* [SD05], um jogo de Gestão de Engenharia de Software, tem como objetivo ensinar como gerir um projeto sem apoio. Esta solução apoia-se na aprendizagem de processos de Gestão através da prática, sem utilizar qualquer tipo de ajuda;
- *iTest Learning* [Far12] é um jogo que serve de apoio ao ensino de testes de Software na sua fase de planeamento. Os tópicos de planeamento abordados são *scope*, itens de teste, tipos de teste, níveis de teste, ferramentas e artefactos;
- *XMED* [WTK08] tem como objetivo auxiliar o ensino de monitorização e controlo de Software. Este jogo é composto por perguntas que devem ser respondidas consoante toda a informação que o jogo fornece;
- *A Ilha dos Requisitos* [TZG10] é um jogo que aborda os Requisitos de Software. É um complemento ao conhecimento desta área, uma vez que não ensina nenhum conceito;

- *SimSoft* [XCY12] depara o jogador com a área de riscos da Gestão de Engenharia de Software. São apresentados conceitos essenciais e o objetivo do jogador é lidar com determinados riscos de um cenário real;
- *O Jogo das 7 Falhas* [DD11] é um Jogo Sérió que simula testes de caixa-preta. O objetivo é descobrir falhas existentes em várias funções de um Software mas, para o jogar, é necessário ter conhecimento básico de testes de Software;
- *iLearnTest* [Rib14] aborda a área dos testes de Software e centra-se em 5 áreas: processo de revisão, análise estática via ferramentas, categorias das técnicas de concepção de testes, técnicas de caixa-preta e técnicas de caixa-branca.

Foi elaborada uma tabela comparativa de vários fatores que pode ser encontrada no capítulo 3, de forma a servir de apoio a pessoas interessadas na Engenharia Informática e desenvolvimento de jogos, ou qualquer pessoa que queira encontrar o mercado de Jogos Sérió de Engenharia de Software existente.

2.3 Padrões Pedagógicos

Um padrão é uma abstração de algo concreto que se repete recorrentemente num contexto não arbitrário, segundo a definição de Riehle e Zullighoven [RZ96]. A utilização de padrões pedagógicos na educação é muito habitual. A finalidade de uma aprendizagem eficaz faz com que seja imprescindível motivar o aprendiz a partir de certos pré requisitos, extraídos do Pedagogical Patterns Project [PPP15].

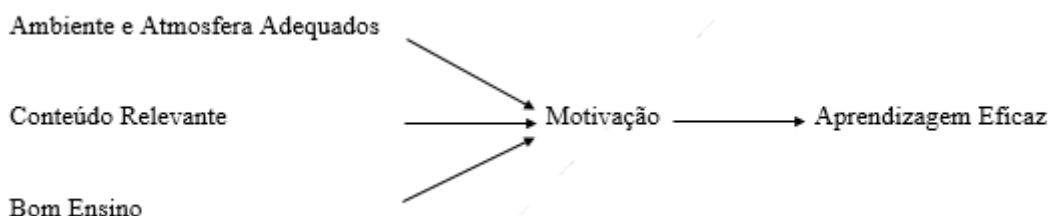


Figura 1 – Pré requisitos para uma aprendizagem eficaz [PPP15]

Para melhorar a eficácia da aprendizagem, foi criada uma lista de funções de aprendizagem e ensino, que refere atividades cognitivas e metacognitivas, cujo objetivo passa pelo seu uso na atividade de ensino, por parte dos educadores. Grösser [Gro07] apoiou-se no

trabalho de Shuell e Moran [SM94], para organizar essas funções de aprendizagem que os dois definiram. Organizou-as em 5 grupos: preparação; manipulação de conhecimento; relações de ordem superior; regulação do aluno; ações produtivas.

1. Preparação:
 - a. Ativação de conhecimento – Relembrar os alunos acerca de temáticas necessárias para a aprendizagem ou perguntar o que já é conhecido sobre o tópico a lecionar;
 - b. Motivação – Estimular a persistência e contribuição do aluno;
 - c. Expectativas – Os alunos devem ter uma ideia geral dos objetivos da aprendizagem em questão;
 - d. Atenção - Focar os alunos na informação relevante;
2. Manipulação do conhecimento:
 - a. Codificação - Ajudar alunos a atribuir significados pessoais à nova informação;
 - b. Comparação - Procurar similaridades e diferenças que permitam fazer relações de ordem superior de compreensão;
 - c. Repetição - Induzir múltiplas perspectivas e fazer revisões sistemáticas;
 - d. Interpretação - Ajudar alunos a converter informação de uma forma de representação para outra;
 - e. Exemplificação - Motivar alunos a esclarecer os conhecimentos, usando novos exemplos;
3. Relações de ordem superior:
 - a. Combinação, Integração, Síntese - Guiar os alunos a combinar os vários fragmentos de informação de forma a permitir a sua integração e síntese. O desenvolvimento de esquemas e tabelas são exemplos de como esta função pode ser ensinada;
 - b. Classificação - Habilitar alunos a determinar categorias de conceitos;
 - c. Sumarização – Conduzir os alunos a escrever resumos que representem a informação;
 - d. Análise – Encaminhar os alunos a dividir o conhecimento em partes e a determinar como estas se relacionam;
4. Regulação do aluno:
 - a. *Feedback* - Alunos têm de interpretar o *feedback* sobre adequação e exatidão da sua compreensão;
 - b. Avaliação - Dar aos alunos a oportunidade de interpretar e avaliar o *feedback*, assim como a oportunidade de avaliar o seu próprio trabalho relativamente a certos critérios;

- c. Monitorização - Dar aos alunos a oportunidade de monitorizar a sua própria aprendizagem, de forma a determinar se existiu um progresso significativo;
 - d. Planeamento - Ajudar alunos em métodos organizacionais para cumprimento de tarefas;
5. Ações produtivas:
- a. Geração de Hipóteses - Encorajar alunos a experimentar diferentes métodos de resolução;
 - b. Dedução - Ajudar alunos a criar as suas conclusões sobre a informação dada;
 - c. Explicação - Guiar alunos na construção mental e uso de modelos de causa-efeito;
 - d. Aplicação - Ensinar alunos a usar métodos para resolução de exercícios e problemas;
 - e. Produção e Construção - Guiar alunos a inventar um produto.

Estas funções são um importante instrumento no ensino e são utilizadas para assistir o estudante a adquirir conhecimento de forma eficiente. Foi importante fazer o seu levantamento para se preceber se são usadas nos jogos educativos e, desta forma, se são cobertas eficazmente e se cumprem os objetivos pedagógicos das áreas.

2.4 Padrões de Desenho para Jogos

Um padrão de desenho de jogos descreve um problema de desenho que é recorrente. Ao definir-se um padrão, estabelece-se a solução para a resolução do problema, como ela deve ser aplicada e em que circunstâncias.

Os padrões de desenho de jogos são cerca de 296 e estão divididos em 11 categorias. Foram Björk e Holopainen [BH04] quem desenvolveu esta coleção de padrões. As 11 categorias são:

- Padrões para elementos de jogo (48 padrões);
- Padrões para recursos e sua gestão (20 padrões);
- Padrões para informação, comunicação e apresentação (20 padrões);
- Padrões para ações e eventos (44 padrões);
- Padrões para estruturas narrativas, previsão e imersão (31 padrões);
- Padrões para interação social (30 padrões);
- Padrões para objetivos (26 padrões);
- Padrões para estruturas objetivo (20 padrões);

- Padrões para sessões de jogo (20 padrões);
- Padrões para mestria do jogo e equilíbrio (27 padrões);
- Padrões para repetibilidade de meta jogos e curvas de aprendizagem (10 padrões).

Estes padrões constituem um jogo e são essenciais no seu desenvolvimento.

Kelle, Klembke e Specht [KKS11], mapearam os padrões de desenho de jogos com as funções de aprendizagem e ensino de Shuell e Moran. Associaram os padrões que se identificavam com cada função. Desse mapeamento surgiram as próximas tabelas.

| Função de Aprendizagem | Classe de Padrões de Desenho |
|-------------------------------|--|
| Ativação de Conhecimento | Padrões para objetivos, exemplo “reconnaissance” |
| Motivação | Padrões para ações e eventos, exemplo “rewards” |
| Atenção | Padrões para elementos de jogo, exemplo “clues” |
| Expectativas | Padrões relacionados com objetivos, exemplo “predefined goals”; padrões de narrativa, exemplo “anticipation” |

Tabela 1 – Mapeamento das funções de aprendizagem de “Preparação” [KKS11]

| Função de Aprendizagem | Classe de Padrões de Desenho |
|-------------------------------|---|
| Codificação | Padrões para informação |
| Comparação | Padrões para informação |
| Repetição | Padrões para meta jogos, exemplo “replayability” e “randomness” |
| Interpretação | Padrões para objetivos, exemplo “gain information” |
| Exemplificação | Padrões para elementos de jogo, exemplo “levels” |

Tabela 2 – Mapeamento das funções de aprendizagem de “Manipulação de conhecimento” [KKS11]

| Função de Aprendizagem | Classe de Padrões de Desenho |
|---------------------------------|--|
| Combinação, Integração, Síntese | Padrões para objetivos, exemplo “gain information” |
| Classificação | Padrões para informação e comunicação, exemplo “perfect information” |
| Sumarização | Padrões para informação e comunicação, exemplo “communication channels”; padrões para sessões de jogo, exemplo “time limits” |
| Análise | Padrões para mestria de jogo e equilíbrio, exemplo “strategic knowledge” |

Tabela 3 – Mapeamento das funções de aprendizagem de “Relações de ordem superior” [KKS11]

| Função de Aprendizagem | Classe de Padrões de Desenho |
|-------------------------------|--|
| Feedback | Padrões para ações e eventos, exemplo “score”; padrões para mestria de jogo e equilíbrio, exemplo “near miss indicators”; padrões para informação, exemplo “progress indicators” |
| Avaliação | Padrões para informação, exemplo “status indicators”; padrões para ações e eventos, exemplo “rewards” e “penalties” |
| Monitorização | Padrões para informação, exemplo “status indicators”; padrões para ações e eventos, exemplo “rewards” e “penalties” |
| Planeamento | Padrões para mestria de jogo e equilíbrio, exemplo “stimulated planning” |

Tabela 4 – Mapeamento das funções de aprendizagem de “Regulação do aluno” [KKS11]

| Função de Aprendizagem | Classe de Padrões de Desenho |
|-------------------------------|--|
| Geração de Hipóteses | Padrões para interação, exemplo “exploração” |
| Dedução | Padrões para estruturas objetivo, exemplo “player defined goals” |
| Explicação | Padrões para informação, exemplo “direct information”; padrões para elementos de jogo, exemplo “clues” e “helpers” |
| Aplicação | Padrões para elementos de jogo, exemplo “clues” |
| Produção e Construção | Padrões de imersão, exemplo “creative control” |

Tabela 5 – Mapeamento das funções de aprendizagem de “Ações produtivas” [KKS11]

Todos os exemplos disponibilizados fazem parte da coleção de padrões referenciada em cima. O exemplo de padrão “clues”, para elementos de jogo, identifica as pistas presentes no jogo.

Este mapeamento tem relevante interesse no desenvolvimento de jogos educativos, uma vez que identificam os padrões responsáveis por cobrir as funções de aprendizagem e ensino. Não é da responsabilidade dos padrões de desenho de jogos abordar os tópicos educativos, mas com este mapeamento percebe-se onde melhor encaixar cada função, ao longo do jogo.

Numa análise feita por Maciuszek, Ladhoff e Martens [MLM11], foram analisados padrões de conteúdo de jogos. O objetivo foi criar um catálogo de padrões para a base de uma história de jogo usual, a ser usada nos Jogos Sérios. O estudo refletiu sobre padrões de desenho para 4 tipos de conteúdo - missão, personagem, ambiente e item - apresentados na seguinte tabela:

| Padrão de Design de Conteúdo | Padrões de desenho | Aplicação |
|-------------------------------------|--|--|
| Missão | 13 padrões de desenho, exemplos: “coming out of the closet”, “declaration of love”, “interrupted wedding/ Runaway bride” | Integração com conteúdo educacional fácil não requerendo estruturas separadas, podendo qualquer dos padrões atuar com caminhos de aprendizagem |
| Personagem | Padrões de desenho de heróis de ação (comum masculino); padrões de desenho de soldado, | Boas personagens necessárias para estabelecer ligação forte |

| | | |
|----------|--|---|
| | “sexy” e frágil (comum feminino); entre outros | |
| Ambiente | 6 padrões de desenho, exemplos: “auction house”, “battleground”, “guilt bank”, “guilt castle”, “housing” e “mailbox” | Ambiente com propósitos colaborativos. Catálogo de padrões de design arquiteturais |
| Item | 5 padrões de desenho de itens, exemplos: “tool”, “reference material”, “map”, “meter” e “souvenir” | Para garantir a interatividade, facilidade de experimentação e variedade de estratégias, os jogos necessitam de itens |

Tabela 6 – Padrões de desenho de conteúdos [MLM11]

2.5 Mapeamento de Padrões de Desenho de Jogos

Num trabalho de investigação realizado anteriormente [Let15][PFL15], foi apresentado um processo de mapeamento dos padrões de desenho de jogos para os objetivos pedagógicos da área da Gestão de Engenharia de Software. O objetivo deste mapeamento centrou-se em identificar os padrões e técnicas de desenho de jogos que melhor serviam para o ensino dos diferentes conteúdos em questão. Esta necessidade adveio do facto de os requisitos para o desenvolvimento de um bom jogo não coincidirem, necessariamente, com os requisitos para uma aprendizagem eficiente.

Neste trabalho de investigação procurou-se uma forma eficaz de mapear os padrões de desenho de jogos para os objetivos pedagógicos dos Requisitos de Software e, portanto, abordou-se este processo utilizado na área da Gestão de Engenharia de Software.

Este mapeamento resultou na pirâmide demonstrada na figura 2, onde estão relacionados três pontos:

- Gestão de Engenharia de S;
- Funções de Ensino e Aprendizagem de Shuell e Moran;
- Padrões de Desenho de Jogos de Björk e Holopainen.

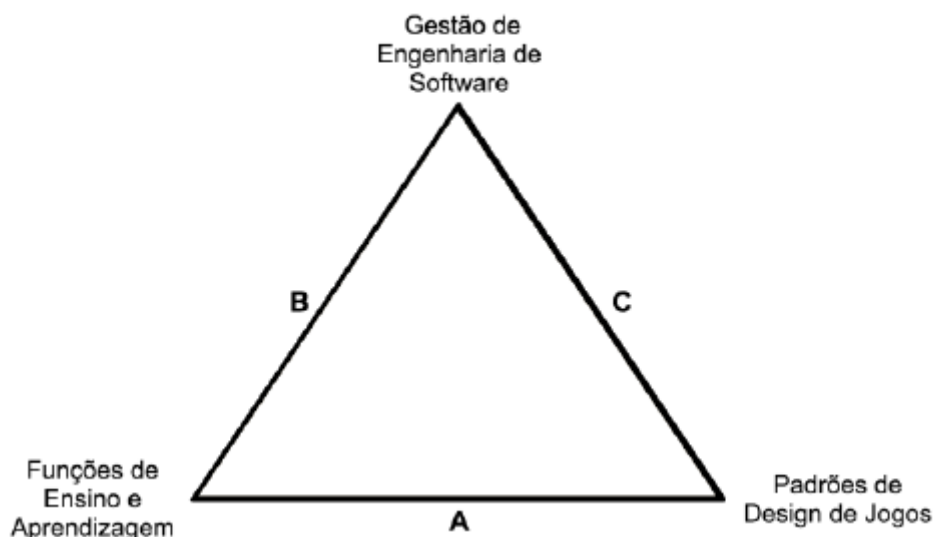


Figura 2 – Abordagem ao mapeamento [Let15]

A aresta A já tinha sido identificada por Kelle [KKS11], apontando os padrões de desenho de jogos que se relacionam com as funções de aprendizagem de Shuell e Moran, como descrito no capítulo anterior.

Para a identificação da aresta B, foi feito um questionário a especialistas da área de Gestão de Engenharia de Software, para reconhecerem as funções de aprendizagem mais importantes em cada tópico da gestão de software. O questionário apresentava uma tabela para relacionar as funções de aprendizagem e ensino com os tópicos de Gestão de Engenharia de Software. O objetivo era determinar quais as funções presentes em cada um dos tópicos. A tabela 7 mostra as funções validadas pelos especialistas, bem como a classe de padrões de desenho de jogos apropriada para a sua implementação.

| Tópico Gestão Engenharia de Software | Função de Aprendizagem | Padrão de Desenho de Jogos |
|---|---|---|
| Iniciação e definição do âmbito | Feedback | Padrões para ações e eventos (ex: score pattern) e padrões para informação (ex: progress indicator pattern) |
| Planeamento do projeto de software | Feedback | Padrões para ações e eventos (ex: score pattern) e padrões para informação (ex: progress indicator pattern) |
| Concretização do projeto de Software | Monitorização | Padrões para informação (ex: status indicator pattern) |
| Revisão e avaliação | Atenção, Interpretação, Análise, Avaliação e Explicação | Padrões para estruturas narrativas (ex: surprises pattern), padrões para objetivos (ex: predefined goals pattern), padrões para mestria do jogo e equilíbrio (ex: randomness pattern), padrões para informação (ex: status indicator) |
| Fecho | Aplicação | Padrões para elementos de jogo |

| | | |
|---|----------------------------------|--|
| Análise do projeto de engenharia de software | Atenção, Análise e Monitorização | Padrões para estruturas narrativas (ex: surprises pattern), padrões para mestria do jogo e equilíbrio (ex: randomness patter) e padrões para informação (ex: status indicator pattern) |
| Ferramentas de gestão de engenharia de software | Aplicação | Padrões para elementos de jogo |

Tabela 7 – Mapeamento dos objetivos pedagógicos de Gestão de Engenharia de Software para padrões de desenho de jogos [Let15]

Por último, a relação entre a Gestão de Engenharia de Software e os padrões de desenho de jogos definiu-se com auxílio de um jogo sério da área, o SimSE. O objetivo deste jogo é educar os jogadores na área da Engenharia de Software, através da simulação do desenvolvimento de um projeto dentro de uma empresa, onde o jogador assume o papel de gestor do projeto. Durante o jogo é possível exercer um conjunto de ações próprias de um gestor de projetos de software, como por exemplo:

- Despedir funcionários;
- Aumentar salários;
- Atribuir bónus;
- Atribuir tarefas a cada elemento da equipa;
- Monitorizar o desempenho da equipa;
- Monitorizar o estado do projeto;
- Comprar ferramentas de desenvolvimento de software;
- Analisar as características físicas e psicológicas de cada membro da equipa;
- Analisar a experiência e conhecimentos de cada membro da equipa;
- Monitorizar o tempo, orçamento e resultado final.

Depois de se analisar o jogo para identificar os padrões de desenho de jogos presentes, desenvolveu-se uma experiência com estudantes, para avaliar a eficácia do jogo no ensino de Gestão de Engenharia de Software. A experiência passou por analisar os conhecimentos que aqueles adquiriram ao jogar o jogo.

Uma turma de 31 estudantes, sem conhecimento prévio em Engenharia de Software, foi dividida em dois grupos uniformes, tendo um dos grupos respondido primeiramente a um questionário de conhecimento de Engenharia de Software, composto por 25 perguntas. Depois os dois grupos jogaram o jogo SimSE e responderam de novo a um inquérito de conhecimento. Com esta experiência percebeu-se que os alunos adquiriram conhecimento ao jogar o jogo. A figura 3 apresenta os resultados dos questionários:

- B1 – Questionário do grupo B antes de jogar o SimSE;
- B2 – Questionário do grupo B depois de jogar o SimSE;
- A – Questionário do grupo A depois de jogar o SimSE.

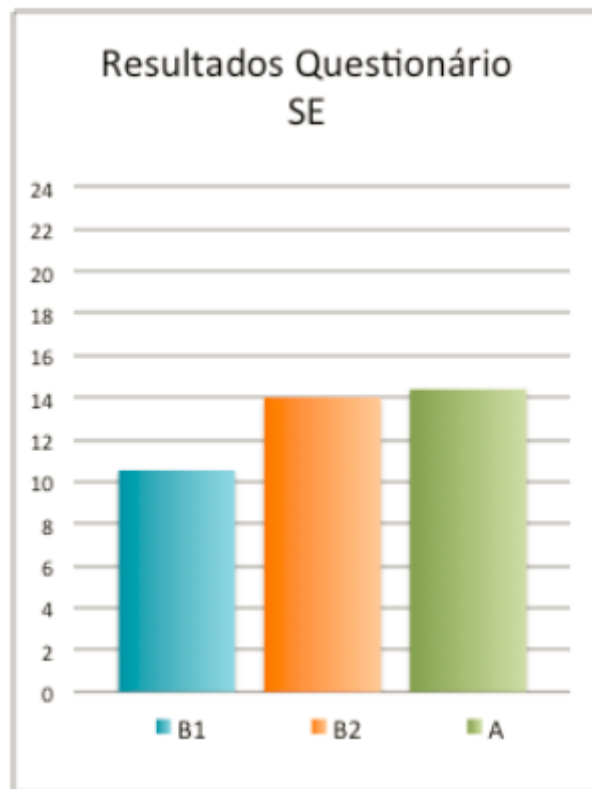


Figura 3 – Resultados dos questionários da experiência [Let15]

2.6 Resumo e Conclusões

Este trabalho de investigação foca-se na aplicação da Engenharia de Software no ensino. Foram explanadas as competências que os jogos digitais podem desenvolver no jogador, bem como as diferenças entre Jogos Sérios e jogos de entretenimento. Realizou-se o levantamento dos Jogos Sérios a ser analisados no capítulo 3 e fez-se uma descrição sucinta para cada um deles.

Abordou-se as funções de aprendizagem e ensino, que são divididas em 5 grupos: a preparação, a manipulação de conhecimento, relações de ordem superior, a regulação do aluno e as ações produtivas.

Foram apresentados os padrões de desenho e os padrões pedagógicos e foi encontrado um mapeamento entre os padrões de desenho e as funções de aprendizagem, bem como entre os padrões de desenho e os padrões de desenho de conteúdos de jogos. Estes mapeamentos permitem

perceber onde as funções de aprendizagem e os padrões de desenho de conteúdos de jogos se encontram, enquanto se joga.

Durante a revisão da literatura concluiu-se que não existe nenhuma tabela comparativa de jogos de Engenharia de Software, tanto para verificar o mercado de jogos e as subáreas em que incidem, como para aquilatar da sua disponibilidade para experimentação.

De igual modo, não se encontrou nenhuma ferramenta que permitisse verificar presença de padrões de desenho de jogos, nos jogos de Engenharia de Software. Com essa análise, seria possível saber quais as funções de aprendizagem que os jogos abordam e se cobrem os objetivos pedagógicos de cada sub área. Desta forma, seria mais prático perceber a aplicabilidade dos jogos ao ensino.

O trabalho de investigação que foi referido neste capítulo aborda um mapeamento interessante para o ensino da Gestão de Engenharia de Software. É importante mapear todas as áreas de forma a fazer-se uma análise crítica aos jogos existentes, para certificar se eles vão de encontro aos objetivos pedagógicos de cada área. Posto isto, depois da análise para a Gestão de Engenharia de Software, ficam em falta as restantes subáreas da Engenharia de Software, sendo o foco deste trabalho os Requisitos de Software.



Capítulo 3

Papel da Engenharia de Software no Ensino

Neste capítulo é apresentada uma comparação dos jogos sérios de Engenharia de Software encontrados, onde são analisados vários tópicos considerados de interesse, bem como os padrões de desenho de jogo existentes nesses jogos que são passíveis de experimentação.

É definido o processo de mapeamento de padrões de desenho de jogos para os objetivos pedagógicos da área dos Requisitos de Software, baseado naqueles definidos para a Gestão de Engenharia de Software, que pode ser encontrado no capítulo 3.

São, ainda, propostas melhorias aos jogos de Gestão de Engenharia de Software e de Requisitos de Software, uma vez que é possível perceber se os objetivos pedagógicos destas áreas são cobertos, através do mapeamento de padrões de desenho de jogos.

3.1 Análise Comparativa de Jogos Sérios de Engenharia de Software

Depois do levantamento de Jogos Sérios, foi decidido analisar vários aspetos considerados pertinentes para a finalidade do trabalho, direcionados, tanto para a Engenharia Informática e desenvolvimentos de jogos, como para o vasto mercado de Jogos Sérios de Engenharia Software. A tabela 8 apresenta os jogos analisados e os tópicos de análise escolhidos.

- Mulp - Multijogador, para perceber se os jogos se focam no ensino a nível coletivo;
- Plat - Plataforma, que mostra se o jogo é desenhado para a *web* ou se é uma aplicação para jogar no *desktop*;

Papel da Engenharia de Software no Ensino

- API, *Application Programming Interface*, se tem ou não uma API disponível;
- Prog - Linguagem de programação usada;
- Graf - Tipo de grafismo, sabendo se o jogo é a duas ou três dimensões;
- Play - Disponível para jogar.

| Games | Theme | Game Type | Mul | Plat | API | Prog | Graf | Play |
|-----------------------|------------------------|-----------|-----|---------|-----|----------------------|------|------|
| SimSE | Management | PC | No | Desktop | Yes | Java | 2D | Yes |
| SE RPG | Management | PC | No | WEB | No | ActionScript (Flash) | 3D | Yes |
| PlayScrum | Management | Cards | Yes | Cards | No | - | - | No |
| SESAM | Management | PC | No | Desktop | No | Ada 95 + TCL/TK | 2D | No |
| AMEISE | Management | PC | Yes | Desktop | No | Ada 95 + TCL/TK | 2D | No |
| Sim JavaSP | Management | PC | No | Desktop | No | Java | 2D | No |
| SimSoft | Management | PC | No | Desktop | No | Java | 2D | No |
| U-Test | Test | PC | No | WEB | No | ActionScript (Flash) | 2D | Yes |
| iTest Learning | Test | PC | No | Desktop | No | Java | 2D | No |
| O Jogo das 7 Falhas | Test | PC | No | WEB | No | Java | 2D | No |
| iLearnTest | Test | PC | No | WEB | Yes | Construct2 | 2D | Yes |
| A Ilha dos Requisitos | Requirements | PC | No | WEB | No | ActionScript (Flash) | 2D | Yes |
| XMED | Monitoring and Control | PC | No | Desktop | Yes | Java | 2D | Yes |

Tabela 8 – Análise de Jogos Sérios de Engenharia de Software

Como se pode verificar, o jogo *PlayScrum* não é jogado no computador mas não deixa de ser um Jogo Sérico. Excluindo este exemplo e o *AMEISE*, nenhum dos outros jogos encontrados suporta multijogador. Alguns jogos analisados permitem a comparação de resultados finais, mas, apesar desta vertente mais social, não foram considerados como sendo *multiplayer*.

Os jogos apresentados são, maioritariamente, de Gestão, mas também foram encontrados exemplos para Testes de Software e Requisitos de Software.

Alguns jogos estavam disponíveis para experimentação e foram analisados no âmbito dos padrões de desenho de jogos. As tabelas 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 e 19 descrevem os

Papel da Engenharia de Software no Ensino

padrões de desenho encontrados em cada jogo. Esses padrões foram descobertos jogando os diferentes jogos. A partir da lista de padrões definida por Björk e Holopainen [BH04] descreveu-se os subpadrões encontrados.

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|---------------------------|--|--|---|---|---|---|--|
| Elementos de jogo | Estes padrões descrevem objetos de jogo que definem a área da realidade de jogo ou que os jogadores podem manipular (48 padrões) | | | | | | |
| Game World | Ambiente onde se joga, determinado pela relação espacial dos elementos do jogo. | Todo o ecrã do jogo, os empregados e as barras de informação, a pontuação e o tempo. | Todo o ecrã do jogo, com os empregados, as barras de informação, a pontuação e o tempo. | Todo o ecrã do jogo, com os avatares, os desafios e exemplos, ajudas, tempo. | Todo o ecrã do jogo, com o mapa da ilha, tempo, avatar, diferentes áreas. | Todo o ecrã do jogo, as perguntas e a área do menu de informação. | Todo o ecrã do jogo, avatar, caixas desafio, pontuação. |
| Reconfigurable Game World | Reconfiguração do ambiente de jogo, incluindo as relações e atributos dos elementos de jogo e as regras de jogo. | Não | Não | Não | Não | Não | Possibilidade de alterar o jogo para qualquer tema definido pelo utilizador. |
| Levels | Parte do jogo onde são cumpridas tarefas até um determinado objetivo. | Não | No modelo em cascata só é possível passar à tarefa seguinte quando a anterior estiver pronta. | Os três artefactos são sequenciais e dentro de cada um, os desafios também só podem ser cumpridos quando o anterior for ultrapassado. | Cada ponto da ilha é um determinado nível. O jogador tem que responder aos diferentes desafios de forma sequencial. | Perguntas sequenciais | Não |
| Inaccessible Areas | Parte do ambiente de jogo onde o jogador não pode entrar. Por pontuação insuficiente ou porque o nível ainda não foi atingido. | Não | Não | O jogador só pode entrar nos novos desafios quando os anteriores forem respondidos. | O jogador só pode entrar em determinadas áreas quando conseguir ultrapassar os desafios de outras. | Como as perguntas são sequenciais o jogador só consegue ver as seguintes quando responder a todas as que estão para trás. | Não |
| Enemies | Unidades ou avatares que dificultam os objetivos ao jogador. | Novos requisitos que aparecem de forma aleatória. | Não | Não | Quando o jogador erra uma pergunta perde alguns segundos de tempo. | Não | Não |
| Avatars | Elemento de jogo a partir do | Não | Personagem escolhida | Personagem escolhida | Personagem | Não | Personagem |

Papel da Engenharia de Software no Ensino

| | | | | | | | |
|------------------|---|-----|---|---|-----|-----|-----|
| | qual o jogador é representado e pode afetar o ambiente de jogo. | | pelo jogador, entre um rapaz e uma rapariga | pelo jogador, com opção de um rapaz ou uma rapariga | | | |
| Clues | Elemento de jogo que informa sobre como atingir os objetivos de jogo. | Sim | Sim | Sim | Sim | Sim | Sim |
| Score | Representação numérica do sucesso do jogador. | Sim | Sim | Sim | Sim | Sim | Sim |
| High Score Lists | Listas de jogadores organizados pelas pontuações. | Não | Não | Sim | Sim | Não | Não |

Tabela 9 – Padrões para elementos de jogo nos Jogos Sérios selecionados

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|--------------------------|--|---|---|--------|-----------------------|------|------------|
| Recursos e gestão | Estes padrões descrevem diferentes tipos de recursos que podem ser controlados pelos jogadores e pelo sistema (20 padrões) | | | | | | |
| Resources | Usados para ativar ações no jogo | Atribuir tarefas aos empregados (desenvolver, rever e corrigir os artefactos) | Atribuir tarefas aos empregados (desenvolver, rever e corrigir os artefactos) | Não | Não | Não | Não |

Tabela 10 – Padrões para recursos e gestão nos Jogos Sérios selecionados

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|---|---|-------|--------|---|---|---|------------|
| Informação, comunicação e apresentação | Estes padrões descrevem como a informação sobre o estado de jogo é tratada, como por exemplo a ocultação de certa informação para avaliação do jogador (20 padrões) | | | | | | |
| Game State Overview | Jogador tem informação geral sobre o estado do jogo. | Sim | Sim | Até ao momento atual. Os níveis acima são escondidos. | Até ao momento atual. Os níveis acima são escondidos. | Até ao momento atual. Os níveis acima são escondidos. | Sim |

Tabela 11 – Padrões para informação, comunicação e apresentação nos Jogos Sérios selecionados

Papel da Engenharia de Software no Ensino

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|------------------------|---|---|---|---|---|---|--|
| Ações e eventos | Estes padrões controlam que tipo de ações estão disponíveis para os jogadores, como estas se relacionam com as alterações do estado do jogo, e como estas se relacionam com os objetivos dos jogadores (44 padrões) | | | | | | |
| Limited Set of Actions | O jogador só pode escolher algumas ações. | Quem faz o quê | Escolha da equipa, das ferramentas, do que fazer e o que usar | Até ao momento que nos encontramos. Os níveis acima são escondidos. | Não | Não | Não |
| Downtime | O jogador não pode afetar o final do jogo durante algum tempo. | Modo pausa. Definir o que fazer e só depois avançar no tempo. | Modo pausa. Definir o que fazer e só depois avançar no tempo. | Não | Não | Não | Não |
| Irreversible Actions | Ações cujo efeito no estado do jogo não pode ser alterado. | Quando se aumenta o salário de um jogador não é possível voltar ao salário anterior | Não | Não | Depois de errar uma resposta o tempo é decrementado. | Ao errar a resposta vamos para a próxima e não se pode repetir para aumentar a pontuação. | Não |
| Penalties | O jogador recebe uma desvantagem por não completar corretamente um requisito do jogo. | Não atinge a pontuação máxima se não cumprir todos os objetivos corretamente | Não atinge a pontuação máxima se não cumprir todos os objetivos corretamente. | Não atinge a pontuação máxima se não cumprir todos os objetivos corretamente. | Há uma penalização no tempo de cada vez que se erra uma resposta. | Não atinge a pontuação máxima se não cumprir todos os objetivos corretamente | Não atinge a pontuação máxima se não cumprir todos os objetivos corretamente |

Tabela 12 – Padrões para ações e eventos nos Jogos Sérios selecionados

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|--|---|-------------------------------------|---|--|---|--|--|
| Estruturas narrativas, previsão e imersão | Estes padrões lidam com o enredo, imersão e empenho sobre o jogo pelos jogadores (31 padrões) | | | | | | |
| Freedom of Choice | O jogador pode fazer escolhas durante o jogo. | Aumentar salário, despedir pessoas. | Comprar ferramentas, despedir pessoas, contratar pessoas. | Não | Não | Não | Escolher qual o desafio a cumprir. |
| Storytelling | O jogo conta um história. | Não | Não | Não | História da ilha dos requisitos. | Não | Não |
| Narrative Structures | Ao longo do jogo lidamos com uma determinada narrativa que forma uma história. | Não | Só podemos avançar para uma atividade quando a anterior | Narrativa de 3 artefactos. Só se pode fazer um de cada vez, por ordem. | A ilha tem vários pontos que vão sendo completados por ordem. | Sequência de perguntas que forma a narrativa | Estrutura de menus que descrevem diferentes capítulos de um corpo de conhecimento. |

Papel da Engenharia de Software no Ensino

| | | | | | | | |
|------------|---|--|--|--|---|-----|-----|
| | | | estiver completa. | | | | |
| Characters | Representações de pessoas ou papéis no jogo. | Personagem principal (manager) que somos nós e empregados. | Personagem principal (manager) que somos nós e empregados. | Personagem principal, pessoa que nos entrevista e ajuda. | Jack, a personagem principal e as várias personagens da ilha. | Não | Não |
| Surpsises | Eventos ou consequências inesperadas que influenciam na atenção do jogador. | Novos requisitos sem aviso prévio. | Não | Não | Não | Não | Não |

Tabela 13 – Padrões para estruturas narrativas, previsão e imersão nos Jogos Sérios selecionados

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|-------------------------|---|-------|--------|--------|-----------------------|------|------------|
| Interação social | Estes padrões mostram como os jogos suportam a interação social entre os jogadores (30 padrões) | | | | | | |
| Single Player | | | | | | | |

Tabela 14 – Padrões para interação social nos Jogos Sérios selecionados

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|------------------|--|--|--|---|---|---|--|
| Objetivos | Padrões que lidam com os objetivos que os jogadores pretendem atingir (26 padrões) | | | | | | |
| Collection | Completar uma série de objetivos | Atingir o máximo de pontuação, seguindo o modelo corretamente, sabendo qual o orçamento e tempo a cumprir. | Atingir o máximo de pontuação, seguindo o modelo corretamente, sabendo qual o orçamento e tempo a cumprir. | Atingir o máximo de pontuação respondendo corretamente aos objetivos. | Atingir o máximo de pontuação respondendo corretamente aos objetivos. | Atingir o máximo de pontuação respondendo corretamente aos objetivos. | Atingir o máximo de pontuação respondendo corretamente aos objetivos. |
| Exploration | Aprender sobre o layout do ambiente do jogo para localizar partes ou objetos. | Não | Não | Não | Não | Não | Conhecer a estrutura de menus com os diferentes objetivos da matéria que se está a aprender. |

Tabela 15 – Padrões para objetivos nos Jogos Sérios selecionados

Papel da Engenharia de Software no Ensino

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|------------------------------|---|--|---|--|--|--|---|
| Estruturas objetivo | Estes padrões descrevem como a jogabilidade afeta os objetivos (20 padrões) | | | | | | |
| Predefined Goals | Hierarquia de objetivos, para tornar o jogo conciso. | Tomar medidas para finalizar o projeto. | Tomar medidas para finalizar o projeto. | Responder adequadamente aos desafios para aumentar a pontuação final. | Responder adequadamente aos desafios para aumentar a pontuação final. | Responder adequadamente aos desafios para aumentar a pontuação final. | Responder adequadamente aos desafios para aumentar a pontuação final. |
| Dynamic Goal Characteristics | Certas características que mudam durante o jogo. | Os requisitos não são pré-definidos. Há requisitos a aparecer a meio do jogo de forma aleatória. | Não | Não | Não | Não | Não |
| Optional Goals | Objetivos que o jogador não tem que cumprir durante o jogo. | O jogador pode entregar o projeto sem ele estar finalizado. | O jogador pode entregar o projeto sem ele estar acabado. | Não | Não | Não | Não |
| Hierarchy of Goals | Os objetivos do jogo formam uma hierarquia. | Não | O jogador só pode passar à fase seguinte quando terminar a fase em questão. | Só se pode fazer um desafio de cada vez e é obrigatório acabar esse desafio. | Só se pode fazer um desafio de cada vez e é obrigatório acabar esse desafio. | Só se pode fazer um desafio de cada vez e é obrigatório acabar esse desafio. | Não |

Tabela 16 – Padrões para estruturas objetivo nos Jogos Sérios selecionados

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|------------------------|--|---|---|--------|-----------------------|------|------------|
| Sessões de jogo | Estes padrões lidam com as características das instâncias e sessões de jogo e as suas limitações, possibilidades e atributos da participação do jogador no jogo (20 padrões) | | | | | | |
| Single-Player Games | Sim | | | | | | |
| Turn-Based Games | São as ações do jogador que fazem mudar o estado do jogo e o seu progresso. | É o jogador que controla o modo de pausa. O jogador avança até ao próximo evento (algum empregado fala, alguma tarefa concluída). | É o jogador que controla o modo de pausa. O jogador avança até ao próximo evento. | Não | Não | Não | Não |

Tabela 17 – Padrões para sessões de jogo nos Jogos Sérios selecionados

Papel da Engenharia de Software no Ensino

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|-------------------------------------|--|--|--|--|--|--|--|
| Mestria do jogo e equilíbrio | Estes padrões descrevem como os jogadores podem usar as suas habilidades e experiência de jogo e como é possível balancear a jogabilidade para jogadores com diferentes habilidades (27 padrões) | | | | | | |
| Empowerment | O jogador pode melhorar o objetivo do jogo. | O objetivo do jogo é fazer a maior pontuação possível. O jogador é responsável por aumentar a pontuação que vai receber. | O objetivo do jogo é fazer a maior pontuação possível. O jogador é responsável por aumentar a pontuação que vai receber. | O objetivo do jogo é fazer a maior pontuação possível. O jogador é responsável por aumentar a pontuação que vai receber. | O objetivo do jogo é fazer a maior pontuação possível. O jogador é responsável por aumentar a pontuação que vai receber. | O objetivo do jogo é fazer a maior pontuação possível. O jogador é responsável por aumentar a pontuação que vai receber. | O objetivo do jogo é fazer a maior pontuação possível. O jogador é responsável por aumentar a pontuação que vai receber. |
| Randomness | Efeitos ou eventos no jogo que não são previsíveis. | Novos requisitos aleatórios. | Não | Não | Não | Não | Não |

Tabela 18 – Padrões para mestria do jogo e equilíbrio nos Jogos Sérios selecionados

| Nome | Descrição | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|--|--|--|--|--------|-----------------------|------|---|
| Repetibilidade de meta jogos e curvas de aprendizagem | Estes padrões lidam com os problemas com vêm de fora da jogabilidade de um só jogador (10 padrões) | | | | | | |
| Replayability | Um jogo tem prespetivas diferentes quando jogado várias vezes | Cada vez que se joga é diferente e acontecem coisas diferentes | Cada vez que se joga é diferente e acontecem coisas diferentes | Não | Não | Não | Pode-se navegar no jogo de maneira diferente de cada vez que se joga. |

Tabela 19 – Padrões para repetibilidade de meta jogos e curvas de aprendizagem nos Jogos Sérios selecionados

A partir desta análise, usou-se a relação de Kelle [KKS11] para mapear os padrões de desenho de jogos apropriados para as funções de ensino e aprendizagem, assim como a respetiva presença em cada jogo. Este mapeamento está identificado nas tabelas 20, 21, 22, 23 e 24, organizadas consoante o tipo de função:

- Preparação;
- Manipulação de conhecimento;
- Relações de ordem superior;

Papel da Engenharia de Software no Ensino

- Regulação do aluno;
- Ações produtivas.

| Função | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|--------------------------|-------|--------|--------|-----------------------|------|------------|
| Ativação de conhecimento | Sim | Sim | Sim | Sim | Sim | Sim |
| Motivação | Sim | Sim | Sim | Sim | Sim | Sim |
| Expectativas | Sim | Sim | Sim | Sim | Sim | Sim |
| Atenção | Sim | Sim | Sim | Sim | Sim | Sim |

Tabela 20 – Funções de aprendizagem de **preparação** usadas nos Jogos Sérios selecionados

| Função | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|----------------|-------|--------|--------|-----------------------|------|------------|
| Codificação | Sim | Sim | Sim | Sim | Sim | Sim |
| Comparação | Sim | Sim | Sim | Sim | Sim | Sim |
| Repetição | Sim | Sim | Não | Não | Não | Sim |
| Interpretação | Sim | Sim | Sim | Sim | Sim | Sim |
| Exemplificação | Sim | Sim | Sim | Sim | Sim | Sim |

Tabela 21 – Funções de aprendizagem de **manipulação de conhecimento** usadas nos Jogos Sérios selecionados

| Função | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|---------------------------------|-------|--------|--------|-----------------------|------|------------|
| Combinação, Integração, Síntese | Sim | Sim | Sim | Sim | Sim | Sim |
| Classificação | Sim | Sim | Sim | Sim | Sim | Sim |
| Sumarização | Sim | Sim | Sim | Sim | Sim | Sim |
| Análise | Sim | Sim | Sim | Sim | Sim | Sim |

Tabela 22 – Funções de aprendizagem de **relações de ordem superior** usadas nos Jogos Sérios selecionados

Papel da Engenharia de Software no Ensino

| Função | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|---------------|-------|--------|--------|-----------------------|------|------------|
| Feedback | Sim | Sim | Sim | Sim | Sim | Sim |
| Avaliação | Sim | Sim | Sim | Sim | Sim | Sim |
| Monitorização | Sim | Sim | Sim | Sim | Sim | Sim |
| Planeamento | Sim | Sim | Sim | Sim | Sim | Sim |

Tabela 23 – Funções de aprendizagem de **regulação do aluno** usadas nos Jogos Sérios selecionados

| Função | SimSE | SE RPG | U-Test | A Ilha dos Requisitos | XMED | iLearnTest |
|-----------------------|-------|--------|--------|-----------------------|------|------------|
| Geração de Hipóteses | Sim | Sim | Não | Não | Não | Não |
| Dedução | Sim | Sim | Sim | Sim | Sim | Sim |
| Explicação | Sim | Sim | Sim | Sim | Sim | Sim |
| Aplicação | Sim | Sim | Sim | Sim | Sim | Sim |
| Produção e Construção | Sim | Sim | Sim | Sim | Sim | Sim |

Tabela 24 – Funções de aprendizagem de **ações produtivas** usadas nos Jogos Sérios selecionados

Como se pode verificar, as funções de repetição e geração de hipóteses não estão presentes em alguns dos jogos.

No caso da repetição: Os jogos *U-Test*, *A Ilha dos Requisitos* e *XMED* são jogos de sequência. Estes três jogos nunca repetem o mesmo padrão, uma vez que não se pode voltar a uma pergunta anterior. O *SimSE* e o *SE RPG* são jogos onde se pode fazer várias vezes a mesma ação, seja aumentar o salário de um funcionário ou atribuir a mesma tarefa duas vezes. Também o *iLearnTest* permite navegar para desafios que já foram cumpridos.

Em relação a geração de hipóteses: Os jogos *SimSE* e *SE RPG* podem ter abordagens diferentes. Ao jogador é dado poder para decidir os elementos do jogo, os empregados ou o modelo para o processo de desenvolvimento de Software. Desta forma, a sua experiência vai ser diferente de jogo para jogo.

Com o levantamento das funções presentes em cada jogo, é possível perceber se os objetivos de ensino de Engenharia de Software são cumpridos em cada um deles.

3.2 Mapeamento de Padrões para o Ensino de Requisitos de Software

O processo usado no capítulo anterior [Let15], para mapear os padrões de desenho de jogos para os objetivos pedagógicos da área da Gestão de Engenharia de Software, foi estendido à área dos Requisitos de Software. A abordagem usada foi a mesma:

1. Relacionar as funções de aprendizagem e ensino de Shuell e Moran com os padrões de desenho de jogos de Björk e Holopainen, já resolvido e apresentado por Kelle (tabelas 1, 2, 3, 4 e 5);
2. Relacionar os Requisitos de Software com as funções de aprendizagem e ensino;
3. Relacionar os Requisitos de Software com os padrões de desenho de jogos.

Relacionar os Requisitos de Software com as funções de aprendizagem e ensino

Para relacionar as funções de aprendizagem e ensino de Shuell e Moran recorreu-se à ajuda de profissionais que trabalham na área dos Requisitos de Software. Os participantes da *Quatic (International Conference on the Quality of Information and Communications Technology)* e da *International Requirements Engineering Conference*, de Engenharia Informática ou Engenharia Eletrotécnica, que tinham trabalhos na área dos Requisitos de Software, foram contactados. Foi-lhes pedido que seleccionassem as funções de aprendizagem e ensino de Shuell e Moran que são usadas em cada tópico dos Requisitos de Software, bem como, a importância de cada função no seu contexto geral dos Requisitos de Software.

Colecionou-se apenas 6 respostas de especialistas, tanto nacionais como internacionais. Na tabela 25 são apresentados os resultados da relação das funções de Shuell e Moran com os tópicos dos Requisitos de Software.

Papel da Engenharia de Software no Ensino

| | Fundamentos dos Requisitos de Software | Processo de Requisitos | Elicitação de Requisitos | Análise de Requisitos | Especificação de Requisitos | Validação de Requisitos | Considerações práticas | Ferramentas de Requisitos de Software |
|---------------------------------|--|------------------------|--------------------------|-----------------------|-----------------------------|-------------------------|------------------------|---------------------------------------|
| Ativação de conhecimento | 33.3% | 50% | 33.3% | 16.7% | 33.3% | 16.7% | 16.7% | 16.7 |
| Motivação | 33.3% | 33.3% | 66.7% | 33.3% | 50% | 33.3% | 16.7% | 0% |
| Expectativas | 83.3% | 83.3% | 83.3% | 83.3% | 83.3% | 50% | 16.7% | 33.3% |
| Atenção | 16.7% | 66.7% | 100% | 50% | 50% | 16.7% | 16.7% | 16.7% |
| Codificação | 0% | 16.7% | 100% | 50% | 83.3% | 16.7% | 16.7% | 16.7% |
| Comparação | 0% | 16.7% | 33.3% | 66.7% | 66.7% | 33.3% | 0% | 0% |
| Repetição | 16.7% | 50% | 50% | 33.3% | 50% | 50% | 50% | 33.3% |
| Interpretação | 0% | 16.7% | 50% | 50% | 66.7% | 33.3% | 16.7% | 16.7% |
| Exemplificação | 16.7% | 66.7% | 100% | 100% | 100% | 83.3% | 50% | 33.3% |
| Combinação, Integração, Síntese | 0% | 16.7% | 33.3% | 33.3% | 83.3% | 66.7% | 16.7% | 33.3% |
| Classificação | 0% | 16.7% | 33.3% | 0% | 83.3% | 50% | 0% | 0% |
| Sumarização | 0% | 0% | 16.7% | 0% | 83.3% | 16.7% | 0% | 16.7% |
| Análise | 0% | 16.7% | 16.7% | 50% | 16.7% | 0% | 0% | 0% |
| Feedback | 0% | 33.3% | 50% | 16.7% | 66.7% | 83.3% | 33.3% | 16.7% |
| Avaliação | 16.7% | 16.7% | 16.7% | 50% | 66.7% | 66.7% | 16.7% | 16.7% |
| Monitorização | 0% | 33.3% | 33.3% | 33.3% | 83.3% | 83.3% | 16.7% | 83.3% |
| Planeamento | 33.3% | 100% | 50% | 33.3% | 33.3% | 33.3% | 33.3% | 16.7% |
| Geração de hipóteses | 0% | 50% | 50% | 50% | 50% | 33.3% | 33.3% | 16.7% |
| Dedução | 0% | 16.7% | 50% | 33.3% | 50% | 33.3% | 33.3% | 16.7% |
| Explicação | 0% | 16.7% | 50% | 33.3% | 83.3% | 33.3% | 16.7% | 16.7% |
| Aplicação | 0% | 83.3% | 83.3% | 83.3% | 83.3% | 100% | 33.3% | 16.7% |
| Produção e construção | 0% | 16.7% | 16.7% | 16.7% | 83.3% | 50% | 83.3% | 16.7% |

Tabela 25 – Resultados obtidos pela análise do inquérito aos especialistas

O objetivo foi reunir as funções de Shuell e Moran, presentes em cada tópico dos Requisitos de Software, com maior percentagem de concordância. Com a análise dos resultados verificou-se que a maior parte dos tópicos tinham, pelo menos, uma função com que os especialistas inquiridos

concordavam. Como três tópicos não reuniram concordância absoluta em nenhuma função de aprendizagem, decidiu-se considerar as concordâncias acima de 80% para esses mesmos tópicos.

Como se pode verificar na tabela 25, depois de se aceitar as margens de 80% para os tópicos que não reuniram concordância de 100%, todos os tópicos conseguem ser cobertos por funções de aprendizagem e ensino.

Na tabela 26, encontram-se os tópicos dos Requisitos de Software e as funções que lhes foram relacionadas. A partir das tabelas 1, 2, 3, 4 e 5, referidas no capítulo 2, relacionou-se os padrões de desenho de jogos mapeados para as funções de aprendizagem, usadas para cobrir os tópicos dos Requisitos de Software.

| Tópico dos Requisitos de Software | Função de Aprendizagem | Padrão de Desenho de Jogos |
|--|--------------------------------------|---|
| Fundamentos dos Requisitos de Software | Expectativas | Padrões relacionados com objetivos, exemplo “predefined goals”; padrões de narrativa, exemplo “anticipation” |
| Processo de Requisitos | Planeamento | Padrões para mestria de jogo e equilíbrio, exemplo “stimulated planning” |
| Elicitação de Requisitos | Atenção, Codificação, Exemplificação | Padrões para elementos de jogo, exemplo “clues” e “levels”; Padrões para informação |
| Análise de Requisitos | Exemplificação | Padrões para elementos de jogo, exemplo “levels” |
| Especificação de Requisitos | Exemplificação | Padrões para elementos de jogo, exemplo “levels” |
| Validação de Requisitos | Aplicação | Padrões para elementos de jogo, exemplo “clues” |
| Considerações Práticas | Produção e construção | Padrões de imersão, exemplo “creative control” |
| Ferramentas de Requisitos de Software | Monitorização | Padrões para informação, exemplo “status indicators”; padrões para ações e eventos, exemplo “rewards” e “penalties” |

Tabela 26 – Mapeamento dos tópicos dos Requisitos de Software para as funções de aprendizagem e ensino de Shuell e Moran e padrões de desenho de jogos

Também se questionou quais as funções de aprendizagem e ensino essenciais para o ensino dos Requisitos de Software, na generalidade. As funções foram pontuadas de 1 a 5, sendo que 1 significa que é pouco usada no ensino dos Requisitos de Software e 5 muito usada. Decidiu-se interpretar como funções essenciais, aquelas cuja pontuação foi maior ou igual a 4. Aquelas que reuniram 100% de concordância por parte dos profissionais, que as avaliaram como essencial, foram:

- Expectativas;
- Codificação;
- Exemplificação;

- Aplicação.

Analisando a tabela 26, vemos que estas funções vão ao encontro daquelas que cobriram os tópicos dos Requisitos de Software, realçando a sua importância.

Relacionar os Requisitos de Software com os padrões de desenho de jogos

Para validar a utilização dos padrões de desenho de jogos utilizou-se o jogo *A Ilha dos Requisitos*, cujos padrões encontrados foram descritos anteriormente, nas tabelas 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 e 19. Como este jogo tinha, efetivamente, padrões de desenho nele presentes, foi desenvolvida uma experiência empírica com estudantes, de forma a validar praticamente o uso dos padrões de desenho dentro do jogo.

Para o desenvolvimento da experiência juntou-se um grupo de estudantes do 2º ano de Engenharia Informática e Computação e do 4º ano de Engenharia Eletrotécnica e de Computadores, que nunca tinham tido nenhuma cadeira específica de Engenharia de Software. Os estudantes foram divididos em dois grupos. De forma a minimizar ameaças à validação dos resultados, os grupos foram uniformizados tendo em conta o número de alunos e a média de curso (tabela 27 e Anexo B).

| Grupo | Número de alunos | Média de curso |
|-------|------------------|----------------|
| A | 8 | 14.1 |
| B | 10 | 14.2 |

Tabela 27 – Dados relativos aos grupos participantes na experiência

Na figura 4 pode-se verificar os passos que se seguiram na experiência, bem como a duração de cada fase.

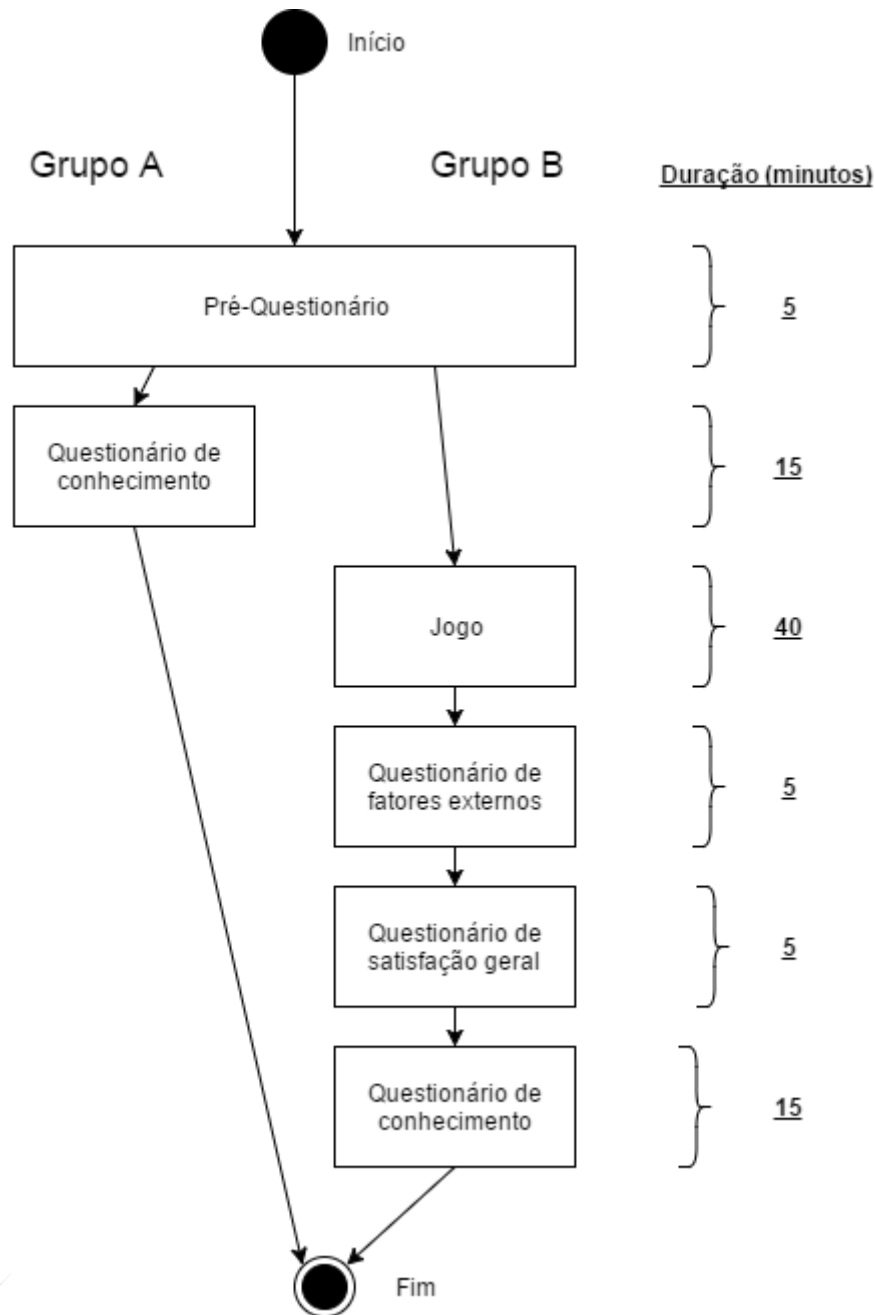


Figura 4 – Delineação da experiência

Ambos os grupos responderam a um pré-questionário (Anexo C), de forma a excluir-se estudantes que já tinham conhecimentos em Engenharia de Software e Requisitos de Software. Um dos elementos do grupo B foi excluído da experiência por ter conhecimentos, tanto em Engenharia de Software, como em Requisitos de Software, tendo os seus resultados sido descartados do estudo.

Papel da Engenharia de Software no Ensino

O pré-questionário consistiu nos seguintes pontos:

1. **Tenho conhecimentos vastos na área da Engenharia de Software;**
2. **Sei o que é um processo de Engenharia de Requisitos;**
3. **Tenho experiência em Engenharia de Requisitos;**
4. **Estou familiarizado com o jogo *A Ilha dos Requisitos*;**
5. **Já joguei o jogo *A Ilha dos Requisitos*.**

Os resultados são apresentados na tabela 28:

| Pergunta | Grupo A (média) | Grupo B (média) |
|----------|-----------------|-----------------|
| 1 | 3 | 2.89 |
| 2 | 2.25 | 1.56 |
| 3 | 1.88 | 1.33 |
| 4 | 1.38 | 1.22 |
| 5 | 1.25 | 1.22 |

Tabela 28 – Resultados do pré-questionário

Todas as perguntas, em ambos os grupos, tiveram valores iguais ou inferiores a 3, como era pretendido. Estes resultados validaram a participação dos estudantes na experiência por, supostamente, não terem conhecimentos em Engenharia de Software nem na sub área dos Requisitos de Software. Segundo os inquiridos, os estudantes não tinham conhecimento do jogo escolhido para a experiência, à excepção do aluno excluído.

Posteriormente, o grupo A respondeu a um questionário de conhecimento (Anexo E) de Engenharia de Software e Requisitos de Software, durante 15 minutos. Este questionário foi elaborado com base no jogo, *A Ilha dos Requisitos*, que continha perguntas direccionadas aos Requisitos de Software. Foram desenvolvidas 10 perguntas, tanto de escolha múltipla, como de opção por verdadeiro/falso. Ambos os grupos responderam a este questionário, o grupo B depois do pré-questionário, o grupo A, depois de jogar o jogo, apresentando-se os resultados obtidos na figura 5.

O grupo A retirou-se da sala para não afetar a continuidade da experiência e o grupo B esteve a jogar o jogo escolhido, durante 40 minutos.

Os elementos do grupo B organizaram-se em pares, sendo informados que iriam jogar um jogo, com a liberdade de poderes pesquisar informação na *internet*, em caso de necessidade. Optou-se por permitir este acesso, porque o jogo é de apoio a estudo e, uma vez que os estudantes não tinham conhecimento prévio, não iriam conseguir responder às perguntas se não tivessem nenhuma ferramenta de pesquisa de informação. Para analisar a viabilidade desta escolha pediu-se aos estudantes para classificarem o uso da internet de 1 (pouco uso) a 5 (muito uso) e a média

foi de 4.44, mostrando que o auxílio da *internet* foi muito importante para o desenrolar da experiência.

Depois de jogar o jogo, os estudantes do grupo B responderam a três questionários:

- Questionário de fatores externos e satisfação geral (Anexo D);
- Questionário de conhecimento, idêntico ao, anteriormente, apresentado ao grupo A.

As perguntas do questionário de fatores externos foram:

1. **Achei que o ambiente da experiência foi apropriado;**
2. **Achei que a duração da experiência foi apropriada;**
3. **Gostei de jogar com o meu colega;**
4. **O ambiente da experiência provocou demasiadas distrações.**

Os alunos responderam a este inquérito durante 5 minutos e os resultados foram os seguintes:

| Pergunta | Média |
|----------|-------|
| 1 | 4.22 |
| 2 | 4.22 |
| 3 | 4.11 |
| 4 | 1.22 |

Tabela 29 – Resultados do questionário sobre fatores externos

Os resultados estiveram dentro do que era pretendido, uma vez que as perguntas 1, 2 e 3 tiveram média superior a 3 e a pergunta 4 apresentou uma média inferior a 3, o que não implicou a exclusão de nenhum estudante.

O questionário de satisfação geral foi respondido de seguida, usando também 5 minutos do tempo da experiência. Foram incluídas as seguintes questões:

1. **No geral, gostei da experiência de jogo;**
2. **A qualidade dos gráficos não constituiu um fator negativo para com a experiência de jogo;**
3. **Achei que esta forma de aprendizagem é interessante;**
4. **Aprendi alguma coisa com a experiência.**

Mais uma vez, os resultados, apresentados na tabela 30, não comprometeram a validade da experiência.

Papel da Engenharia de Software no Ensino

| Pergunta | Média |
|----------|-------|
| 1 | 3.33 |
| 2 | 4.22 |
| 3 | 4 |
| 4 | 4 |

Tabela 30 – Resultados do questionário de satisfação geral

Como as médias deste inquérito estiveram dentro dos padrões pretendidos, nenhum estudante foi excluído.

Por último, foram reservados 15 minutos para o grupo B responder ao questionário de conhecimento, já efetuado pelo grupo A, dando-se depois por terminada a experiência.

O questionário de conhecimento e os seus resultados discriminados encontram-se no Anexo E e nas tabelas 36 e 37. Na tabela 31 apresenta-se a pontuação final dos estudantes, que varia de 0 a 10 e que é a soma dos pontos obtidos, em cada pergunta, por cada estudante.

| Grupo A | Grupo B |
|---------|---------|
| 6 | 6 |
| 2 | 8 |
| 6 | 6 |
| 2 | 6 |
| 5 | 8 |
| 5 | 8 |
| 6 | 9 |
| 3 | 8 |
| | 4 |

Tabela 31 – Resultados obtidos pelo grupo A e grupo B no questionário de conhecimento

Na figura 5 são apresentadas as médias ponderadas dos dois grupos participantes na experiência: o grupo A, que não jogou o jogo, e o grupo B, que respondeu ao inquérito depois de jogar *A Ilha dos Requisitos* durante 40 minutos.

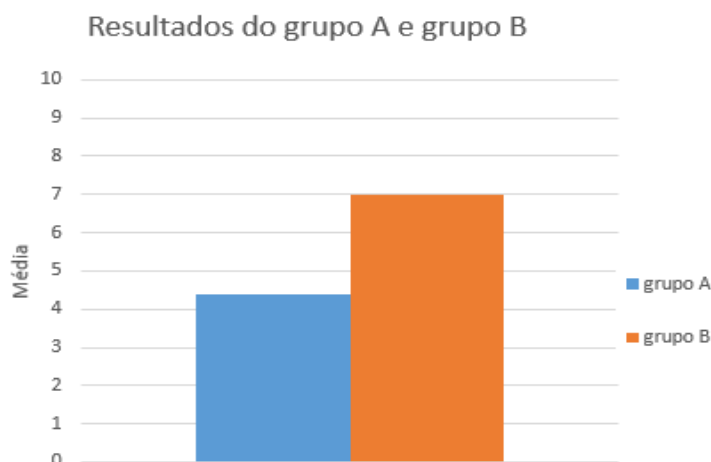


Figura 5 – Resultados do questionário do grupo A (não jogou o jogo) e do grupo B (jogou o jogo antes de responder)

Como se pode verificar, os alunos que jogaram o jogo durante 40 minutos, antes de responder ao questionário, tiveram melhores resultados. Estes resultados traduzem-se numa melhoria de 26.3%. Assim, no grupo A, três alunos registaram nota “negativa”, enquanto que, no grupo B, apenas um aluno não ultrapassou os 5/10 valores; enquanto no grupo A ninguém ultrapassou os 6/10 valores, no grupo B, dos nove elementos validados, quatro tiveram pontuação inferior ou igual a 6/10, perfazendo um total de 44.4%, tendo os restantes cinco obtido resultados superiores ou iguais a 7/10.

O objetivo desta experiência foi validar o uso de um jogo sério, contendo padrões de desenho de jogos, no ensino dos Requisitos de Software e, a partir resultados obtidos, é possível concluir que os estudantes adquiriram aprendizagens sobre a área em questão. Apesar de os grupos serem constituídos por elementos diferentes, o que podia influenciar muito a forma como responderiam ao inquérito, decidiu-se não apresentar o questionário ao grupo B, antes do jogo, para não interferir com a maneira como os estudantes o jogariam. Posto isto, acredita-se que *A Ilha dos Requisitos* proporcionou a aquisição de conhecimento e, conseqüentemente, o uso dos padrões de desenho presentes no jogo, pode ser validado como uma ferramenta para o ensino dos Requisitos de Software.

3.3 Proposta de Melhorias a Jogos Sérios de Engenharia de Software

O mapeamento dos padrões de desenho para os objetivos pedagógicos da Gestão de Engenharia de Software [PL15] e dos Requisitos de Software permite perceber se os Jogos Sérios encontrados estão desenhados de forma a cumprir os objetivos de ensino que se perseguem.

Quando se analisou o mapeamento para a Gestão de Engenharia de Software, os resultados do inquérito aos especialistas mostraram que todos os tópicos da área têm, pelo menos, uma função de Shuell e Moran responsável pelo seu ensino,

Da análise dos resultados obteve-se o seguinte:

- Iniciação e Definição do âmbito – *Feedback*;
- Planeamento do projeto de Software – *Feedback*;
- Concretização do projeto de Software – Monitorização;
- Revisão e avaliação – Atenção, Interpretação, Análise, Avaliação, Explicação;
- Fecho – Aplicação;
- Análise do projeto de Engenharia de Software – Atenção, Análise, Monitorização;
- Ferramentas de gestão de Engenharia de Software – Aplicação.

Pela análise aos jogos realizada neste capítulo, tanto o *SimSE* como o *SE RPG*, ambos jogos de Gestão de Engenharia de Software, constatou-se que têm todas aquelas funções de ensino e aprendizagem presentes. Desta forma podemos dizer que estes dois jogos abordam os objetivos pedagógicos desta área da Engenharia de Software.

Por outro lado, à medida que se joga o jogo, *SimSE*, foi possível perceber certos pontos suscetíveis de serem reformulados, daí terem surgido algumas propostas de melhoria. Visto que os objetivos pedagógicos estavam cobertos pelos padrões de desenho de jogos que se relacionam com as funções de aprendizagem e ensino, estas melhorias foram direcionadas para a jogabilidade e funcionalidade do jogo.

Neste contexto, para o jogo *SimSE* foram propostas as seguintes melhorias:

- **Clarificar as instruções do jogo.** Falta a explicação de funcionamento do modelo de desenvolvimento de software escolhido para jogar;
- **Fornecer mais informação sobre o jogo ao longo do tempo.** Era importante perceber como é que a pontuação do jogo funciona, quando é que se ganha e perde pontos, para aprender dessa forma a gerir o projeto da maneira mais adequada;

- **Sinalizar as mensagens importantes.** Os avisos são mostrados todos da mesma forma. Quando um empregado vai descansar, essa informação é disponibilizada do mesmo modo de um novo requisito que aparece no jogo. Nessas condições pode-se perder informações importantes;
- **Apoiar o jogador durante o processo de gestão.** O jogo dá demasiada autonomia ao jogador, sem o ajudar nas suas decisões. Visto ser um jogo de aprendizagem, àquele devia ser disponibilizado algum protocolo de gestão ou transmitidas indicações de como jogar, ao longo do tempo;
- **Avisar quando o jogador se afasta do modelo escolhido.** Quando se utiliza o modelo em cascata, pode-se jogar livremente, sendo os erros apenas refletidos na pontuação final. O jogador deveria ser avisado caso começasse uma tarefa fora da ordem correta;
- **Disponibilizar dicas sobre o jogo.** Se um empregado merece um aumento ou fica parado durante muito tempo, deveria aparecer alguma notificação. A gestão dos recursos humanos poderia assim ser abordada de uma forma mais completa.

Apesar dos tópicos da Engenharia de Software estarem todos cobertos, é impotente definir estas melhorias para se cumprir os objetivos pedagógicos da melhor forma.

O jogo *SE RPG*, muito parecido com o *SimSE*, apresenta também quase todos os pontos fracos acima referidos. O único ponto já contemplado no jogo, é a aplicação do modelo de desenvolvimento de software como história de jogo, não permitindo que o jogador se afaste dele. Quando se joga o jogo com o modelo em cascata, só é possível passar à próxima fase do projeto quando a fase anterior está terminada. O jogo usa um tipo de narrativa para manter o seguimento do modelo. Apesar deste problema identificado no *SimSE* ter sido colmatado, há ainda muitos fatores que podem ser melhorados. Já a visualização geral do jogo piorou no *SE RPG*, uma vez que se pode navegar por três salas dentro do jogo, ao contrário do *SimSE*, onde tudo se centra no mesmo ecrã.

Quanto ao mapeamento dos Requisitos de Software da tabela 26, conseguiu-se apurar que o jogo analisado *A Ilha dos Requisitos* também engloba todas as funções de aprendizagem e ensino necessárias para cobrir todos os tópicos desta área, não sendo necessário propôr nenhuma melhoria neste aspeto.

Da mesma maneira que se propôs melhorias aos dois jogos de Gestão de Engenharia de Software, também *A Ilha dos Requisitos* tem pontos que podem ser aperfeiçoados em termos de jogabilidade:

- **Iniciar o jogo com o ensino de alguns tópicos.** Todos os desafios abordam temas que não foram antes referenciados, o que faz com que os jogadores tenham que pesquisar os seus significados fora do ambiente de jogo. Desta forma o jogo passa a servir como apoio à aprendizagem dos Requisitos de Software, que não é o seu objetivo;
- **Incluir maior variedade de conceitos.** O jogo pode ser alargado a mais desafios, para abordar mais conceitos;

- **Abordar o mesmo tópico em desafios diferentes**, para aprofundar o conhecimento e relacioná-lo com outras áreas;
- **Sistematizar a história do jogo**. O jogador tem que procurar qual o próximo passo a seguir e num jogo onde o tempo é muito importante, como este, seria mais rentável optar por uma hierarquia de desafios mais visível;
- **Reduzir a penalização de errar um desafio**. Como este jogo não aborda a matéria que se apresenta nos desafios, o jogador vai responder de forma insistente à pergunta, até acertar. Como há penalização de tempo, sempre que se submete um desafio que esteja errado, considera-se que esta técnica de tentativa e erro não é adequada;
- **Melhorar o grafismo do jogo**, que não se mostrou inteiramente positivo na experiência empírica com os alunos.

Estas melhorias interferem também com a cobertura dos tópicos dos Requisitos de Software, uma vez que o jogo se torna mais claro e coerente.

Tendo em conta que não há mapeamento de padrões de desenho de jogos para as outras áreas da Engenharia de Software, decidiu-se não propôr melhorias aos jogos que abordam as outras áreas, sem saber, antecipadamente, se os tópicos são cobertos pelos padrões de desenho de jogos.

3.4 Resumo e conclusões

Neste capítulo efetuou-se uma análise comparativa dos Jogos Sérios de Engenharia de Software existentes. Avaliou-se os jogos segundo 6 critérios:

- Tema;
- Tipo de jogo;
- Existência de multijogador;
- Plataforma;
- API acessível;
- Linguagem de programação;
- Tipo de grafismo;
- Disponibilidade para experimentar.

Também se avaliou a presença de padrões de desenho de jogos definidos por Björk e Holopainen, naqueles possíveis de se experimentar. Os jogos avaliados foram o *SimSE*, *SE RPG*, *U-Test*, *A Ilha dos Requisitos*, *XMED* e *iLearnTest*.

Papel da Engenharia de Software no Ensino

Foi abordado o processo de mapeamento, descrito no capítulo 2, agora para os Requisitos de Software. A abordagem de relacionamento de padrões de desenho de jogos para os objetivos pedagógicos da área escolhida, foi descrita neste capítulo. Realizou-se um questionário a profissionais da área dos Requisitos de Software para validar a importância das vinte e duas funções de aprendizagem e ensino de, para o ensino dos diferentes tópicos desta área. Também se desenvolveu uma experiência empírica com estudantes sem conhecimentos em Engenharia de Software, para validar o uso dos padrões de desenho de jogos num jogo de Requisitos de Software.

Avaliou-se os jogos existentes para Engenharia de Software e propuseram-se melhorias para a cobertura dos objetivos pedagógicos e para uma otimização do jogo, em termos de jogabilidade.

Capítulo 4

Conclusões e Trabalho Futuro

A Engenharia de Software tem uma grande variedade de conceitos, que faz com que o seu ensino tenha de ser muito distinto nas diferentes áreas, como os Requisitos de Software ou a Gestão de Software. Na abordagem de cada área é necessário adotar técnicas de ensino específicas devido aos diferentes tópicos e objetivos educacionais.

O uso de jogos como ferramenta educativa é cada vez mais procurado, porque incentiva a aprendizagem através de atividades lúdicas e atrativas, onde os níveis de motivação se mantêm sempre elevados. O mercado dos Jogos Sérios está em constante crescimento e a Engenharia de Software, naturalmente, acompanha essa tendência. Contudo, apesar de existirem jogos versando algumas das suas áreas, o seu desenvolvimento não segue nenhum processo específico de desenho de jogos. Desta forma, torna-se bastante difícil perceber se os objetivos educacionais são cumpridos. A criação de um mapa dos padrões de desenho de jogos para a área da Gestão de Engenharia de Software mostrou-se útil para promover e facilitar o desenvolvimento de novos jogos, no entanto, apenas essa área apresenta, para já, o mapeamento.

Esta dissertação começa por introduzir o contexto do trabalho, o problema apresentado e a motivação, bem como os objetivos que se pretendem alcançar.

Posteriormente apresenta-se a revisão bibliográfica, que se centra no ensino da Engenharia de Software e das suas áreas de conhecimento, no uso de Jogos Sérios para o ensino, nos padrões pedagógicos existentes, no uso de padrões de desenho para jogos e do mapeamento de padrões de desenho de jogos para a Gestão de Engenharia de Software.

Na terceira parte deste trabalho pode-se encontrar uma análise comparativa de Jogos Sérios para a Engenharia de Software, bem como uma análise aprofundada aos padrões de desenho de jogos presentes em cada jogo disponível para ser experimentado. Também se valida o mapeamento para a área dos Requisitos de Software, de padrões de desenho de jogos, encontrado

anteriormente, para a Gestão de Engenharia de Software. São propostas melhorias a jogos sérios, tanto de Gestão de Engenharia de Software como de Requisitos de Software, a nível da cobertura dos objetivos educacionais e de jogabilidade.

4.1 Satisfação dos Objetivos

O uso de Jogos Sérios está a tornar-se cada vez mais frequente na área da Engenharia de Software e os jogos encontrados foram analisados de acordo com uma seleção de diferentes critérios, considerados de interesse para a Engenharia Informática ou desenvolvimento de jogos. Percebeu-se a utilidade em agregar todos os jogos existentes num catálogo único, uma vez que são cada vez mais utilizados e há necessidade de conhecer o que existe no mercado, antes de se desenvolver um novo jogo. Por outro lado, também há mais facilidade em encontrar os jogos e aceder às suas especificidades, nomeadamente, saber que área abordam, se são passíveis de experimentação, em que linguagem de programação estão escritos, entre outros fatores.

Os jogos disponíveis para jogar foram avaliados em termos dos padrões de desenho de jogos neles presentes. Essa análise é útil para indicar se os objetivos pedagógicos são cobertos, através do mapeamento para Gestão de Software e Requisitos de Software encontrado. A avaliação da presença dos padrões de desenho de jogos também auxilia o desenvolvimento de novos jogos, porque torna possível a perceção das falhas de um jogo, proporcionando a sua reformulação nas novas construções.

O mapeamento de padrões de desenho de jogos mostrou-se de relevante interesse para a Gestão de Engenharia de Software, proporcionando uma série de vantagens no desenvolvimento de jogos, de modo que se decidiu validar esse processo, também, para a área dos Requisitos de Software. A criação de um mapa associativo entre os tópicos dos Requisitos de Software e os padrões de desenho de jogos adequados para a sua implementação, permitiu validar todo o processo de mapeamento, também para esta área.

4.2 Trabalho Futuro

Uma vez que o mapeamento de padrões de desenho de jogos para Gestão de Software foi validado para os Requisitos de Software, pode ser regularizado no sentido da sua adequabilidade para as outras áreas. No entanto, apesar deste mapa semântico mostrar resultados interessantes, é importante reformular dois aspetos:

- Aumentar o número de validações das funções de aprendizagem e ensino para cada tópico da área a ser estudada, por parte dos especialistas;
- Alargar a escala da experiência empírica, para se validar os conhecimentos adquiridos a partir dos Jogos Sérios que usam os padrões de desenho de jogos.

Conclusões e Trabalho Futuro

É determinante perceber se o inquérito aos profissionais das áreas é a melhor abordagem ao problema, ou se é possível obter validação das funções de aprendizagem e ensino recorrendo a outra forma de estudo.

A comparação de jogos de Engenharia de Software deve ser alargada a mais jogos que possam vir a existir, bem como a análise de padrões de desenho de jogos, com a disponibilidade de mais jogos passíveis de ser experimentados.

Com o mapeamento para as outras áreas da Engenharia de Software, também se torna mais fácil propôr melhorias aos jogos que nelas se enquadram. A grande finalidade desta proposta é fazer com que os jogos cubram todos os objetivos pedagógicos que cada área se propõe conseguir.

Conclusões e Trabalho Futuro



Referências

- [Abt70] Abt, C. (1970). *Serious Games*. USA, Viking Press.
- [ACM04] ACM/IEEE (2004). *Software Engineering Curriculum*. Guidelines for Undergraduate Degree Programs in Software Engineering.
- [BF14] Bourque, P. and R.E. Fairley (2014). *Guide to the Software Engineering – Body of Knowledge, IEEE Computer Society*. www.swebok.org [march 1, 2016].
- [BH04] Björk, S., Holopainen, J. (2004) *Patterns in Game Design*, Charles River Media, Boston, MA.
- [BM08] Benitti, F., Molléri, J. (2008). *Utilização de um RPG no Ensino de Gerenciamento e Processo de Desenvolvimento de Software*.
- [Cor06] Corti, K. (2006) *Games-based Learning; a serious business application*. PIXEL Learning Limited.
- [DAJ11] Djaouti, D., Alvarez, J., Jessel, J. (2011). *Classifying Serious Games: the G/P/S model*. IRIT – University of Toulouse, France.
- [DD11] Diniz, L. L., Dazzi, R. L. S. (2011) *Jogo Digital para o Apoio ao Ensino do Teste de Caixa-Preta*. In: X Simpósio Brasileiro de Qualidade de Software, Curitiba.
- [DJ00] Drappa, A., Jochen, L. (2000). *Simulation in software engineering training*. Proceedings of the 22nd International Conference on Software Engineering. Pp. 199-208.
- [Far12] Farias, V., Moreira, C., Coutinho, E., Santos, I. (2012). *iTest Learning: Um Jogo para o Ensino do Planejamento de Testes de Software*. Universidade Federal do Ceará.
- [FS10] Fernandes, J., Sousa, S. (2010). *PlayScrum – A Card Game to Learn the Scrum Agile Method*. Universidade do Minho.
- [Gro07] Grösser, M. (2007) ‘Effective teaching: linking teaching to learning functions’, *South African Journal of Education*, Vol. 27, No. 1, pp. 37-52.
- [KKS11] Kelle, S., Klemke, R., Specht, M. (2011) ‘Design patterns for learning games’, *Int. J. Technology Enhanced Learning*, Vol. 3, No. 6, pp. 555-569.

Referências

- [Let15] Letra, P. (2015) *Game Design Techniques for Software Engineering Management Education*. Dissertação de mestrado da Faculdade de Engenharia da Universidade do Porto.
- [Mit03] Mittermeir, R. T., Hochmuller, E., Bollin, A., Jager, S., Nusser, M. (2003) *AMEISE – A Media Education Initiative for Software Engineering*. Concepts, the Environment and Initial Experiences.
- [MLM11] Maciuszek, D., Ladhoff, S., Martens, A. (2011) *Content Design Patterns for Game-Based Learning*. IJGBL 1(3):65-82.
- [MM95] Macaulay, M. and Mylopoulos, J. (1995). *Requirements Engineering: An Educational Dilemma*. Automated Software Engineering 2, 343-351.
- [MS04] Mitchell, A. and Savill-Smith, C. (2004) *The use of computer and video games for learning: A review of the literature*. Learning and Skills Development Agency.
- [Nav06] Navarro, E. (2006). *SimSE: A Software Engineering Simulation Environment*, University of California, Irvine.
- [Oh02] Oh, E. (2002) *Teaching Software Engineering Through Simulation*. University of California, Irvine.
- [PFL15] Paiva, A., Flores, N., Letra, P. (2015) *Towards Game-Based Software Engineering Management Education*. Conference on Higher Education Advances – HEAD, published in Procedia: Social and Behavioral Sciences.
- [PPP15] *The Pedagogical Patterns Project* (2015) www.pedagogicalpatterns.org [march 1, 2016]
- [Pre01] Prensky, M. *Fun, Play and Games: What Makes Games Engaging*, In: Digital Game-Based Learning, 2001.
- [Rib14] Ribeiro, T. (2014). *iLearnTest: Jogo Educativo para Aprendizagem de Testes de Software*. Dissertação de Mestrado. Faculdade de Engenharia da Universidade do Porto.
- [RZ96] Riehle, D. and Zullighoven, H. (1996). *Understanding and Using Patterns in Software Development*. University of Hamburg, Germany.
- [Saw02] Sawyer, B. *Serious Games: Improving Public Policy through Game-based Learning and Simulation*. USA, Woodrow Wilson International Center for Scholars.
- [SD05] Shaw, K., Dermoudy, J. (2005) *Engendering an Empathy for Software Engineering*. School of Computing. University of Tasmania.
- [Sil10] Silva, A. C. (2010). *Jogo Educacional para Apoiar o Ensino de Técnicas para Elaboração de Testes de Unidade*. Dissertação de Curso de Mestrado, Computação Aplicada, UNIVALI, São José.
- [SJB07] Susi, T., Johannesson, M., Backlund, P. (2007) *Serious Games – An overview*, University of Sjøvde (Technical Report HS-IKI-TR-07-001), Skovde, Sweden.

Referências

- [SM94] Shuell, T.J. and Moran, K.A. (1994) 'Learning theories: historical overview and trends', in T. Husen e T.N. Postlethwaite (Eds.): *The International Encyclopedia of Education*, 2nd ed., pp.3340-3345, Pergamon, Oxford, UK.
- [TZG10] Thiry, M., Zoucas, A., Gonçalves, R. (2010). *Promovendo a Aprendizagem de Engenharia de Requisitos de Software através de um Jogo Educativo*. SBIE, Brasil.
- [WTK08] Gresse Von Wagenheim, C., Thiry, M., Kochanski, D. (2008). *Empirical Evaluation of an Educational Game on Software Measurement*. Empirical Software Engineering.
- [XCY12] Xia, J., Caulfield, C., Yeo, D. (2012). *Simsoft: A game for teaching project risk management*. Teaching and Learning Forum.

Referências



Anexo A

Inquérito a Profissionais de Requisitos de Software

O inquérito enviado ao engenheiros profissionais em Requisitos de Software encontra-se anexado nas seguintes páginas.

Questionnaire on Software Engineering Requirements - Learning and Teaching functions

* Required

- Preparation -

Reminding students of prerequisite information or asking oneself what is already known about the topic being learned (Prior knowledge activation)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Learner persistence and contribution need to be nurtured (Motivation)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Learners need to have a general idea of what is to be accomplished from the learning task (Expectations)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Enabling learners to focus on relevant information, disregarding the irrelevant information (Attention)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

- Knowledge Manipulation -

Assisting learners to add personal meaning to new information (Encoding)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Making comparisons in searching for similarities and differences that permit the formation of higher-order relationships characteristic of understanding (Comparison)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

The inducement of multiple perspectives and engaging in systematic reviews are two ways in which this function can be initiated (Repetition)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Assisting learners in converting information from one form of representation to another (Interpreting)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Motivating learners to illustrate by making use of new examples (Exemplifying)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

- Higher Order Relationships -

Isolated pieces of information must be combined in ways that permit integration and synthesis. Developing organizational schemes such as tables and diagrams are examples of how this function can be initiated (Combination, integration, synthesis)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Enabling learners to determine categories of concepts (Classifying)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Guiding learners in writing short statements that represent information (Summarizing)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Guiding learners to break material into constituent parts and to determine how the parts are related (Analyzing)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

- Learner Regulation -

Learners need to interpret feedback on the adequacy and accuracy of their understanding (Feedback)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Providing learners with the opportunity to interpret and evaluate the feedback, as well as the opportunity to evaluate their own work against set criteria and standards (Evaluation)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Providing learners with the opportunity to monitor their own learning progress, to determine if reason able progress is being made (Monitoring)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| | | | | |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1 | 2 | 3 | 4 | 5 |
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Assisting learners in devising methods for accomplish tasks (Planning)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

- Productive Actions -

Encouraging learners to try alternate courses of action or generating alternative solutions (Hypothesis generation)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Assisting learners to draw conclusions from presented information (Inferring)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Guiding learners in constructing mentally and using cause-and-effect models (Explaining)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Teaching learners how to use procedures to perform exercises or solve problems (Applying)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Guiding learners to create a product (Producing and constructing)

Fill the field, by evaluating it from 0 (less important) to 5 (more important), based on their usage in Requirements Engineering. *

| 1 | 2 | 3 | 4 | 5 |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Mark the corresponding sub-areas where the functions are present. *

- None
- Software Requirements Fundamentals
- Requirements Process
- Requirements Elicitation
- Requirements Analysis
- Requirements Specification
- Requirements Validation
- Practical Considerations
- Software Requirements Tools

Can you give an example in the context of requirements engineering?

Your answer

Observations

Your answer

Name

not required

Your answer

University

not required

Your answer

Thank you for your colaboration

Anexo B

Médias dos Participantes na Experiência

Encontra-se anexada a tabela da média de classificação escolar de todos os alunos.

| | Grupo A | Grupo B |
|-----------------------|----------------|----------------|
| | 16.61 | 14.48 |
| | 16.74 | 16.01 |
| | 13.23 | 15.46 |
| | 13.68 | 15.24 |
| | 17.28 | 15.36 |
| | 11.95 | 16.48 |
| | 11.63 | 12.66 |
| | 11.36 | 11.02 |
| | | 11.49 |
| | | 13.57 |
| Média do grupo | 14.1 | 14.2 |

Tabela 32 – Médias dos estudantes que participaram na experiência

Anexo C

Pré-Questionário da Experiência

Nas próximas páginas encontra-se anexado o pré-questionário respondido por todos os alunos que participaram na experiência, bem como os seus resultados.

Os valores que não pertencem ao intervalo que se considera aceitável encontram-se marcados a vermelho nas tabelas de resultados.

Experiência Empírica com Estudantes

TENFOGS06

Maio de 2016

Pré-Questionário

Obrigada, desde já, pela participação nesta experiência.

Antes de começar, solicitávamos a sua colaboração neste breve questionário, com o objetivo de aferir o seu perfil enquanto participante.

Para cada pergunta, pedíamos que respondesse conforme o seguinte: **1 (Discordo plenamente)**, **2 (Discordo parcialmente)**, **3 (Não concordo nem discordo)**, **4 (Concordo parcialmente)**, **5 (Concordo plenamente)**. Deve colocar um “X” na opção com a qual melhor se identifica.

ID: _____

Questionário

| 1. Tenho conhecimentos vastos na área de Engenharia de Software. | | | | | |
|--|--|--|--|--|--|
| 2. Sei o que é um processo de engenharia de requisitos. | | | | | |
| 3. Tenho experiência em engenharia de requisitos. | | | | | |
| 4. Estou familiarizado com o jogo <i>A Ilha dos Requisitos</i> . | | | | | |
| 5. Já joguei o jogo <i>A Ilha dos Requisitos</i> . | | | | | |

Obrigada pelo seu tempo.

| | Perguntas | | | | |
|-----------|------------------|----------|----------|----------|----------|
| | 1 | 2 | 3 | 4 | 5 |
| Q1 | 2 | 2 | 2 | 2 | 1 |
| Q2 | 4 | 3 | 3 | 1 | 1 |
| Q3 | 3 | 2 | 1 | 1 | 1 |
| Q4 | 2 | 1 | 1 | 1 | 1 |
| Q5 | 4 | 3 | 3 | 1 | 1 |
| Q6 | 3 | 1 | 1 | 1 | 1 |
| Q7 | 2 | 3 | 1 | 1 | 1 |
| Q8 | 4 | 3 | 3 | 3 | 3 |

Tabela 33 – Resultados do pré-questionário do grupo A

| | Perguntas | | | | |
|------------|------------------|----------|----------|----------|----------|
| | 1 | 2 | 3 | 4 | 5 |
| Q1 | 3 | 1 | 1 | 1 | 1 |
| Q2 | 2 | 1 | 1 | 1 | 1 |
| Q3 | 3 | 1 | 1 | 1 | 1 |
| Q4 | 3 | 3 | 1 | 1 | 1 |
| Q5 | 4 | 3 | 3 | 1 | 1 |
| Q6 | 2 | 1 | 1 | 1 | 1 |
| Q7 | 3 | 1 | 1 | 1 | 1 |
| Q8 | 3 | 2 | 2 | 3 | 3 |
| Q9 | 3 | 1 | 1 | 1 | 1 |
| Q10 | 3 | 1 | 1 | 1 | 1 |

Tabela 34 – Resultados do pré-questionário do grupo B

Anexo D

Questionário de Fatores Externos e Satisfação Geral da Experiência

Apresenta-se neste anexo o questionário de fatores externos e satisfação geral da experiência, bem como os seus resultados. Este questionário foi respondido apenas pelo grupo B, que jogou o jogo.

Os valores que não pertencem ao intervalo que se considera aceitável encontram-se marcados a vermelho nas tabelas de resultados.

Os valores marcados a amarelo não são considerados válidos devido a questões anteriores.

Experiência Empírica com Estudantes

TENFOGS06

Maio de 2016

Questionário sobre a experiência

Obrigada pela sua participação. Pedíamos, agora, que respondesse a este breve questionário sobre o ambiente da experiência realizada.

Cada pergunta está relacionada com fatores externos à experiência e da sua percepção aos mesmos.

Para cada pergunta, pedíamos que respondesse conforme o seguinte: **1 (Discordo plenamente)**, **2 (Discordo parcialmente)**, **3 (Não concordo nem discordo)**, **4 (Concordo parcialmente)**, **5 (Concordo plenamente)**. Deve colocar um “X” na opção com a qual melhor se identifica.

ID: _____

Questionário

Fatores Externos

| 1. Achei que o ambiente da experiência foi apropriado | | | | | |
|---|--|--|--|--|--|
| 2. Achei que a duração da experiência foi apropriada | | | | | |
| 3. Gostei de jogar com o meu colega | | | | | |
| 4. O ambiente da experiência provocou demasiadas distrações | | | | | |

Satisfação Geral

| 1. No geral, gostei da experiência de jogo | | | | | |
|--|--|--|--|--|--|
| 2. A qualidade dos gráficos não constituiu um factor negativo para com a experiência de jogo | | | | | |
| 3. Achei que esta forma de aprendizagem é interessante | | | | | |
| 4. Aprendi alguma coisa com a experiência | | | | | |

Caso tenha algum comentário ou observação, por favor, registe-o abaixo

Obrigada pelo seu tempo.

| | Perguntas | | | |
|------------|-----------|---|---|---|
| | 1 | 2 | 3 | 4 |
| Q1 | 4 | 4 | 5 | 1 |
| Q2 | 4 | 5 | 4 | 2 |
| Q3 | 4 | 3 | 4 | 1 |
| Q4 | 5 | 5 | 5 | 1 |
| Q5 | 4 | 3 | 3 | 1 |
| Q6 | 5 | 5 | 5 | 1 |
| Q7 | 3 | 4 | 3 | 1 |
| Q8 | 5 | 5 | 5 | 1 |
| Q9 | 5 | 5 | 5 | 1 |
| Q10 | 4 | 4 | 3 | 2 |

Tabela 35 – Resultados do questionário de fatores externos do grupo B

| | Perguntas | | | |
|-----|-----------|---|---|---|
| | 1 | 2 | 3 | 4 |
| Q1 | 4 | 3 | 3 | 3 |
| Q2 | 4 | 4 | 4 | 4 |
| Q3 | 3 | 4 | 4 | 5 |
| Q4 | 3 | 5 | 3 | 3 |
| Q5 | 4 | 4 | 4 | 4 |
| Q6 | 4 | 5 | 5 | 4 |
| Q7 | 3 | 5 | 3 | 4 |
| Q8 | 5 | 5 | 4 | 4 |
| Q9 | 2 | 4 | 5 | 5 |
| Q10 | 3 | 4 | 5 | 4 |

Tabela 36 – Resultados do questionário de satisfação geral do grupo B

Anexo E

Questionário sobre Conhecimentos em Engenharia de Software

Apresenta-se neste anexo o questionário feito aos alunos sobre conhecimentos de ES. Há um total de 10 perguntas e os resultados também são aqui apresentados.

Os valores que não pertencem ao intervalo que se considera aceitável encontram-se marcados a vermelho nas tabelas de resultados.

Os valores marcados a amarelo não são considerados válidos devido a questões anteriores.

Experiência Empírica com Estudantes

TENFOGS06

Maio de 2016

Questionário sobre conhecimentos em Engenharia de Software

Obrigada pela sua participação. Pedíamos, agora, que respondesse a este questionário sobre Engenharia de Software e a sua sub-área da Engenharia de Requisitos

As questões apresentadas podem ser de escolha múltipla ou de Verdadeiro/Falso. Pedíamos que respondesse às questões de acordo com os seus conhecimentos. Deve colocar um CÍRCULO na opção que considera correta. Caso queira corrigir a sua resposta, rique (-) e marque (ou não) outra resposta.

ID: _____

Questionário

1. Entre as seguintes afirmações, assinale aquela que define o termo "requisito".

- a) A diferença entre o estado atual e o estado desejado.
- b) Uma instrução sobre como um requisito deve ser executado.
- c) Uma capacidade que o sistema tem.
- d) Um problema que tenha sido identificado.
- e) Uma característica que o sistema deve apresentar.

2. Qual a ordem correta das fases da Engenharia de Requisitos?

- a) Elicitação - Especificação - Análise - Validação - Verificação – Gestão.
- b) Elicitação - Análise - Especificação - Verificação - Validação – Gestão.
- c) Elicitação - Gestão - Especificação - Análise - Verificação – Validação.
- d) Gestão - Especificação - Análise - Verificação - Validação – Elicitação.
- e) Elicitação - Especificação - Análise - Gestão - Verificação – Validação.

3. O que é a especificação de requisitos?

- a) Descrever o que o software deve fazer, a partir do que foi analisado anteriormente.
- b) Avaliar se os requisitos atendem a critérios objetivos.
- c) Avaliar pontos vagos e resolver os seus conflitos.
- d) Garantir que os requisitos especificados estão corretos, de um ponto de vista externo.
- e) Característica, atributo, habilidade ou qualidade que um sistema deve necessariamente apresentar para ser útil aos seus utilizadores.

4. O que é a verificação de requisitos?

- a) Avaliar se os requisitos representam realmente as necessidades do cliente.
- b) Avaliar pontos vagos.
- c) Avaliar se os requisitos atendem a critérios objetivos e consistentes.
- d) Avaliar as propriedades de um sistema ou produto.
- e) Avaliar a ambiguidade dos requisitos para estes serem vagos.

5. Um analista de requisitos deve elaborar testes funcionais.

- a) Verdadeiro.
- b) Falso.

6. Um analista de requisitos deve mediar discussões e conflitos.

- a) Verdadeiro.
- b) Falso.

7. Um analista de requisitos não precisa de ser bom orador nem um bom ouvinte.

- a) Verdadeiro.
- b) Falso.

8. O que são requisitos funcionais?

- a) Restrições que o software deve atender ou qualidades que o software deve ter.
- b) Funcionalidades do sistema.
- c) Descrição, em termos do negócio, do que deve ser entregue e conseguido para criar valor para o cliente.
- d) Todas as anteriores.
- e) Nenhuma das anteriores.

9. Qual(ais) das seguintes atividades é fundamental nos processos de software?

- a) Validação e Verificação.
- b) Especificação.
- c) Negociação.
- d) Todas as anteriores.
- e) Nenhuma das anteriores.

10. Quais os tipos de requisitos existentes na Engenharia de Software.

- a) Requisitos do projeto.
- b) Requisitos funcionais e não funcionais.
- c) Requisitos do projeto e requisitos do produto.
- d) Todas as anteriores.
- e) Nenhuma das anteriores.

Caso tenha algum comentário ou observação, por favor, registre-o abaixo

Obrigada pelo seu tempo.

| | Perguntas | | | | | | | | | | Pontuação |
|--------------|-----------|---|---|---|---|---|---|---|---|-------|-----------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Q1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 6 |
| Q2 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 |
| Q3 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 6 |
| Q4 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 2 |
| Q5 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 5 |
| Q6 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 5 |
| Q7 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 6 |
| Q8 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 3 |
| Média | | | | | | | | | | 4.375 | |

Tabela 37 – Resultados do questionário de conhecimento do grupo A

| | Perguntas | | | | | | | | | | Pontuação |
|--------------|-----------|---|---|---|---|---|---|---|---|----|-----------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | |
| Q1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 6 |
| Q2 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 8 |
| Q3 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 6 |
| Q4 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 6 |
| Q5 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 8 |
| Q6 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 8 |
| Q7 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 9 |
| Q8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 7 |
| Q9 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 8 |
| Q10 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 4 |
| Média | | | | | | | | | | 7 | |

Tabela 38 – Resultados do questionário de conhecimento do grupo B