

Agile O-MaSE: Framework Ágil Orientado a Agentes*

Fabiana Ribeiro Ferraz Gominho¹, Rosa Maria Esteves Moreira da Costa¹, Vera Maria B. Werneck¹

¹Programa de Mestrado em Ciências Computacionais, Universidade do Estado do Rio de Janeiro (UERJ), Brasil

fgominho@gmail.com, {rcosta, vera}@ime.uerj.br

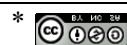
Abstract. Agile methods have been proposed to support the development of multiagent system and currently, some methodologies already incorporate this concept. However, they consider only one specific agile approach. The objective of this paper is to present an agile extension incorporated into the O-MaSE method of developing multiagent systems. The framework Agile O-MaSE was built to be used in modern iterative and incremental projects, replacing the traditional cascade approach. Agile O-MaSE maintained characteristics of the standard O-MaSE methodology, adding activities that characterize and control agile iterations, with a set of activities, artifacts, and agile practices. After defining Agile O-MaSE, a pilot study was carried out with the development of an educational medical game geared towards agents. Preliminary results show positive aspects in the composition of Agile O-MaSE's activities.

Resumo. Métodos ágeis têm sido propostos para apoiar o desenvolvimento de sistema multiagentes e atualmente, algumas metodologias já incorporam esse conceito. Entretanto, elas consideram apenas uma única abordagem ágil específica. O objetivo deste artigo é apresentar uma extensão ágil, incorporada ao método O-MaSE de desenvolvimento de sistemas multiagentes. O framework Agile O-MaSE foi construído para ser utilizado em projetos iterativos e incrementais modernos, substituindo a tradicional abordagem cascata. O Agile O-MaSE manteve as características da metodologia padrão O-MaSE, adicionando atividades que caracterizam e controlam as iterações ágeis, com um conjunto de atividades, artefatos e práticas ágeis. Após a definição da Agile O-MaSE, foi realizado um estudo piloto com o desenvolvimento de um jogo médico educacional orientado a agentes. Resultados preliminares apontam aspectos positivos na composição das atividades da Agile O-MaSE.

1. Introdução

A disseminação de métodos ágeis no desenvolvimento de *software* tem sido bastante aplicada, provocando uma revolução na literatura, na área profissional, e assim, aumentando a busca dos profissionais por conhecimento e atualização sobre *frameworks* ágeis.

Porém, os *frameworks* ágeis já existentes, simplificam os artefatos de modelagem, mas não são tão eficientes em projetos de sistemas multiagentes (MAS - *MultiAgent System*), devido ao nível de complexidade exigido na modelagem de atividades inerentes a esses sistemas (Ferreira et al, 2015).



* O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilha Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

No âmbito dos sistemas multiagentes, ao longo dos anos, foram adaptadas metodologias orientadas a agentes de processos tradicionais (González-Moreno et al, 2014), (Chella et al, 2006), esses métodos adaptados introduzem um único método ágil e conservam vários artefatos, que no futuro podem ainda ser condensados.

Motivada pela disseminação das metodologias ágeis e pela proposta de adaptação em metodologia orientadas a agentes, esse trabalho colabora com a transformação de um método tradicional orientado a agentes, O-MaSE (*Organization - based Multiagent Systems Engineering*), para uma abordagem ágil, denominada: *Agile O-MaSE*. A metodologia O-MaSE foi escolhida, por ter uma proposta de uso interativo e incremental.

Além da introdução, este artigo é composto por mais 5 seções. A seção 2 aborda uma visão geral do desenvolvimento ágil e das abordagens ágeis encontradas em trabalhos correlatos sobre métodos orientados a agentes. A seção 3 introduz o método O-MaSE. A seção 4 apresenta a proposta do *framework Agile O-MaSE*. A seção 5 apresenta a contextualização e o projeto do estudo piloto. Por fim, a seção 6 discute as considerações finais.

2. Desenvolvimento Ágil de Software Orientado a Agentes

O processo de desenvolvimento de software precisava evoluir rapidamente para reduzir as despesas e atender às necessidades de requisitos de negócios em constante mudanças. Em geral, baseados em uma abordagem iterativa, a especificação, a implementação e os testes são intercalados e os resultados do processo de desenvolvimento de software são decididos por meio de um processo de negociação.

Segundo Beck et al (2001), fevereiro de 2001 é o marco inicial do Manifesto para o Desenvolvimento Ágil (MDA), criado por um grupo de 17 pensadores denominados de “a aliança ágil”. O documento contém termos e conceitos gerais, e passou a guiar o modo ágil de gerenciar projetos. O trabalho realizado chegou aos valores descritos na figura 1.



Figura 1 – Valores do Manifesto de Desenvolvimento Ágil de Software (Beck et al, 2001)

Além dos valores do MDA, também foram criados 12 princípios que guiam o desenvolvimento ágil de software, tais como: satisfação do usuário com entregas adiantadas e contínuas, aceitação de mudanças nos requisitos, pessoas do negócio envolvidas no desenvolvimento, equipe motivada, conversas “cara a cara”, intervalos regulares de revisão.

* O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilhamento 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Vários métodos e processos foram propostos para desenvolvimento ágil se destacando o Extreme Programming (XP) e Scrum. O método XP propõe: a criação de histórias de usuário, uso de cartões representativos de classes do sistema (CRC - Class-Responsability-Collaborator), programação em pares e desenvolvimento orientado a testes (TDD - *Test-Driven Development*).

O framework Scrum compartimenta o trabalho em *Sprints* (*time-box* de duração de 1 a 4 semanas), identifica o *Product Backlog* (uma lista de funcionalidades desejadas para o sistema) que é definido pelo papel do *Product Owner*. O papel do *Scrum Master* é garantir que o processo Scrum seja seguido e orientar a equipe no uso efetivo do Scrum. Durante a *Sprint* um incremento de produto é liberado e uma nova *Sprint* é iniciada imediatamente após a conclusão da *Sprint* anterior. As *Sprints* contêm eventos de planejamento, reuniões diárias, o trabalho de desenvolvimento, uma revisão da *Sprint* e uma retrospectiva da *Sprint*.

Um mapeamento sistemático sobre as propostas de desenvolvimento ágil orientado a agentes, identificou 11 trabalhos: (Novo *et al.*, 2018), (Wang, 2018) (Tenso *et al.*, 2017), (Babar *et al.*, 2017), (Seghöfer *et al.*, 2015), (Ferreira *et al.*, 2015), (Noori e Kazemifard, 2015) (Azawi *et al.*, 2014), (Yu *et al.*, 2014). Foram identificadas as metodologias AgilePAssi (Chella, *et al.*, 2006) e INGENIAS-Scrum (GONZÁLEZ-MORENO. *et al.*, 2014). A Agile-Passi incorpora o método XP e INGENIAS-Scrum adapta o processo Scrum na metodologia INGENIAS.

Steghöfer et al (2015) apresentam o método PosoMAS, que é um processo para sistemas multiagentes abertos e auto-organizados, com o ciclo de vida ágil iterativo-incremental do Scrum. Mostra como a nova metodologia foi aplicada em um projeto e as lições aprendidas e por fim, exibe um quadro comparando o PosoMas com outras metodologias de engenharia de software orientada a agentes ágeis.

O trabalho apresentado por Tenso et al (2017) afirma que o artefato história de usuário, utilizado em métodos ágeis, não é o suficiente para entender o panorama geral dos requisitos do sistema, mesmo com a existência de métodos que tentam resolver esse problema, e propõem um novo método de modelagem orientado a agentes ágeis (AAOM).

O trabalho apresentado por Azawi et al (2014) realiza uma investigação nos métodos de desenvolvimento de jogos e fornece um novo método de desenvolvimento de jogos baseado em modelos de desenvolvimento preditivo e adaptativo criando uma cooperação com engenharia de software orientada a agente (AOSE), que introduz uma metodologia híbrida denominada Metodologia de Desenvolvimento de Jogos Ágil para Agentes (AAGDM).

3. Método O-MaSE (Organization - based Multiagent Systems Engineering)

Segundo DeLoach e Garcia-Ojeda (2014) o O-MaSE foi projetado do zero, ou seja, não partiu de outra metodologia, com um conjunto de fragmentos que poderiam ser montados pelos desenvolvedores para atender aos requisitos específicos de seu projeto. O-MaSE não se compromete com um conjunto pré-definido de Fases, em vez disso, define explicitamente Atividades e Tarefas conforme a figura 2 e permite que engenheiros de métodos organizem as atividades em diferentes caminhos baseados na necessidade do projeto.

A metodologia tem sido utilizada para suportar abordagens iterativas e incrementais modernas, bem como abordagens muito mais simples baseadas em cascata.



Entity	Task	Work Product	Role
Requirements Gathering	Requirements Specification	Requirements Spec	Requirements Engineer
Problem Analysis	Model Goals Refine Goals Model Domain	Goal Model Domain Model	Goal Modeler Domain Modeler
Solution Analysis	Model Organization Interfaces Model Roles Define Roles Define Role Goals	Organization Model Role Model Role Description Document Role Goal Model	Organization Modeler Role Modeler
Architecture Design	Model Agent Classes Model Protocols Model Policies	Agent Class Model Protocol Model Policy Model	Agent Class Modeler Protocol Modeler Policy Modeler
Low Level Design	Model Plans Model Capabilities Model Actions	Agent Plan Model Capabilities Model Action Model	Plan Modeler Capabilities Modeler Action Modeler
Code Generation	Generate Code	Source code	Programmer

Figura 2 – Visão geral do O-MaSE (DeLoach e Garcia-Ojeda, 2014)

Ainda segundo DeLoach e Garcia-Ojeda (2014) o fato do O-MaSE não se comprometer com nenhum conjunto específico de Fases, causava um problema por não ter um modelo documentado para descrição de seus processos de design (DPDT - Design Process Documentation Template). Para minimizar este problema, na literatura foi assumido uma abordagem tradicional em cascata, definidas três fases principais: Análise de Requisitos, Projeto e Implementação, conforme a figura 3.

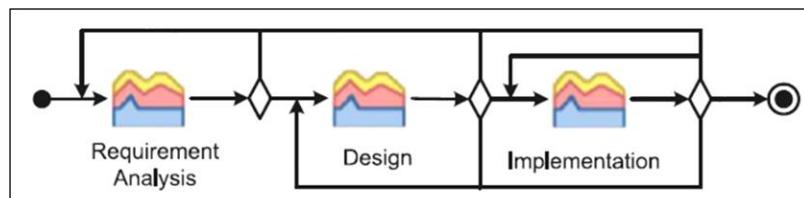


Figura 3 – O-MaSE usando fases em cascata (DeLoach e Garcia-Ojeda, 2014)

Ao utilizar o O-MaSE em um projeto real, o *designer* de processo é livre para definir seu próprio conjunto de Fases e Iterações e atribuir Atividades e Tarefas, conforme apropriado.

4. Proposta *Framework Agile* O-MaSE

Na análise das metodologias ágeis orientadas a agentes encontradas, *Agile-Passi* e *INGENIAS-Scrum*, observou-se que elas adaptaram apenas uma proposta ágil ao método utilizado, *Ingenias* adaptou do *Scrum* e *Passi* adaptou do *XP*, não combinando diversos aspectos, que são propostos em ambos *frameworks* de desenvolvimento ágeis. Contudo, na proposta ágil: *framework Agile* O-MaSE, foram mantidas as Fases da metodologia tradicional



da literatura do O-MaSE, com suas Atividades e Tarefas, e foram adicionados fragmentos dos *frameworks* ágeis: Scrum e XP.

Fazendo referência ao MDA, também se destacam o seguinte valor e princípios para construção do *framework Agile* O-MaSE:

- (i) “Software em funcionamento, mais que documentação abrangente” (2º Valor do MDA);
- (ii) “Entregar Software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos” (3º Princípio do MDA);
- (iii) “A entrega de Software funcional é a medida primária de progresso” (7º Princípio do MDA).

Na **Fase de Análise e Planejamento dos Requisitos**, foram inseridos os produtos: backlog do produto, backlog da iteração e o plano de release originários do Scrum, onde:

Na **Etapa de Levantamento dos Requisitos**, são construídos os Backlogs do Produto e da Iteração; na Etapa de Análise do Problema, são realizadas a Modelagem e Refinamento das Metas e o Plano de Release; na etapa Análise da Solução, são modeladas a interface com a organização e os papéis.

A definição dos requisitos constrói o *Backlog* do Produto com todos os itens que devem ser desenvolvidos para construção do Software. Deve conter requisitos funcionais e não funcionais, e pode ser atualizado a qualquer momento inclusive com os bugs que forem encontrados com o cliente, para acompanhamento das correções. O *Backlog* do Produto deve conter as seguintes informações: item, descrição, tema, prioridade, tipo, estimativa de complexidade, iteração e situação.

O *Backlog* da Iteração, é construído de acordo com a meta/objetivo da iteração, onde o time de desenvolvimento escolhe os requisitos do *backlog* do produto que serão desenvolvidos na iteração corrente. O *Backlog* da Iteração deve conter as informações: meta da iteração, item, história do usuário, tarefas da estória, duração das tarefas, distribuição das tarefas e situação da tarefa.

O Plano de *Release*, é um plano gerencial e deve conter as informações: definição do critério de sucesso da *release*, estimativa dos itens do *backlog* do produto, definição do tamanho e quantidade de iterações, estimativa da velocidade, e associação dos itens do *backlog* do produto as iterações.

Além dos produtos, faltava uma fase importante ligada a qualidade do software, que fizesse referência as características do MDA já destacadas. Dentre os *frameworks* ágeis escolhidos, o XP, se destaca por apresentar uma fase de testes consistente com TDD.

No TDD, o código é desenvolvido de forma incremental, em conjunto com um teste para esse incremento, onde o próximo incremento não é construído até que o desenvolvimento do incremento atual passe no teste. O desenvolvimento de código e os testes são intercalados a cada iteração, conforme a figura 3 (Sommerville, 2019), (Jeffries e Melnik, 2007).



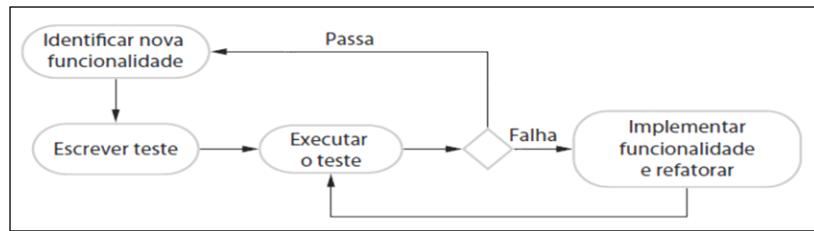


Figura 3 – Fluxo do processo fundamental de TDD (Sommerville, 2019)

Na nova fase, **Fase de Qualidade**, são realizadas duas Atividades: Casos de Testes e Testes.

Na Atividade de Casos de Testes, os testes são escritos a cada iteração, seguindo as estórias selecionadas no *backlog* da iteração.

Na Atividade de Testes, é realizada a cada funcionalidade desenvolvida e só as funcionalidades que estiverem prontas serão liberadas na entrega da iteração, incluindo os testes. As funcionalidades que não passarem nos testes e que não tiveram tempo de ser corrigidas dentro do *time-box* da iteração, devem ser atualizadas na situação da estória no *backlog* da iteração, como erro, para que ela seja corrigida e entregue na próxima iteração.

Visando as boas práticas ágeis, foram adicionados ao *framework Agile* O-MaSE os seguintes eventos do *framework* Scrum: revisão da Sprint (reunião de entrega) e retrospectiva da sprint (reunião de revisão da iteração) e para finalizar foi adicionado o Modelo Adaptativo que é clássico nos métodos ágeis, para formalizar as iterações e incrementos, como mostra a figura 4.

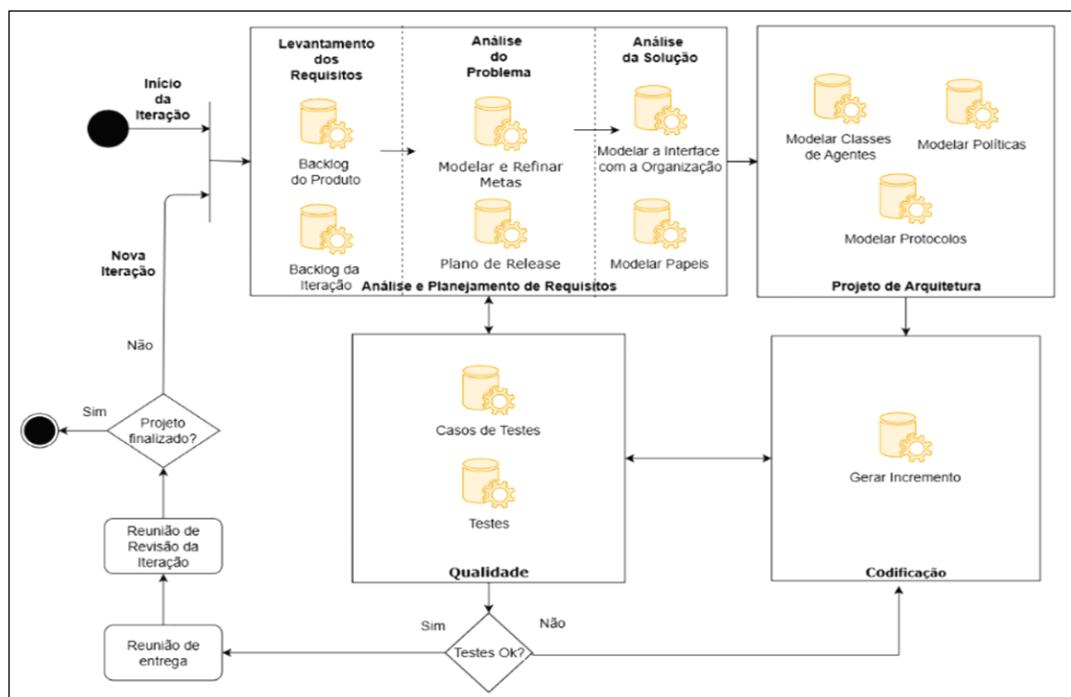


Figura 4 – Framework Agile O-MaSE

Para realizar a construção da proposta piloto do *Agile O-MaSE*, foram considerados: (i) Valores e princípios do Manifesto de Desenvolvimento Ágil (MDA); (ii) Fragmentos dos frameworks XP e Scrum, como parte do processo (descrevendo o que deve ser feito no fragmento específico); (iii) Produtos a serem entregues (artefatos como plano de release, backlogs e casos de testes); (iv) Algumas pré-condições (como os dados de entrada necessários ou a condição de guarda que deve ser verificada antes de iniciar o trabalho especificado no fragmento); (v) Conceitos relacionados a modelos ágeis definidos para serem executados no fragmento de método especificado; (vi) Diretriz (es) que ilustra como aplicar o fragmento e as melhores práticas relacionadas a isso.

5. Estudo Piloto

Esta seção faz uma abordagem de um estudo piloto, com desenvolvimento de um jogo médico educacional orientado a agentes, realizado com uma turma de inteligência artificial (IA) do curso de Ciência da Computação numa Universidade pública do Estado do Rio de Janeiro, com destaque da proposta: *framework Agile O-MaSE*.

O projeto foi dividido em três iterações num *time-box* de 2 semanas, no período de 10 de outubro a 05 de dezembro de 2019. O grupo, composto por quatro alunos, recebeu o cronograma, os artefatos ágeis (*backlog* do produto, *backlog* da iteração e o plano de *release*) e a documentação da metodologia O-MaSE. A mestrandona em ciências computacionais fez a apresentação da metodologia ágil. A Figura 5 apresenta o cronograma.

Id	EDT	Nome da tarefa	Inicio	Término	Nomes dos recursos
1	1	Projeto de Desenvolvimento do Jogo MedEduc - Metodologia Agile O-MaSE	Qui 17/10/19	Qui 28/11/19	Grupo E; Grupo F
2	1.1	Iteração 1	Qui 17/10/19	Qui 31/10/19	
3	1.1.1	Fase de Análise de Requisitos			
4	1.1.1.1	Atividade Planejar Release: Plano da Release			
5	1.1.1.2	Atividade Levantamento dos Requisitos: Backlog do Produto			
6	1.1.1.3	Atividade Análise do Problema: Modelar e Refinar Metas (Diagrama) e Backlog da Iteração			
7	1.1.1.4	Atividade Análise da Solução: Modelo de Interface com a Organização (Diagrama) e Modelo de Papéis (Diagrama)			
8	1.1.2	Fase de Qualidade			
9	1.1.2.1	Casos de Testes			
10	1.1.3	Fase Projeto de Arquitetura			
11	1.1.3.1	Modelar Classes de Agentes, Modelar Políticas e Modelar Protocolos			
12	1.1.4	Fase de Codificação	Qui 24/10/19	Qui 24/10/19	
13	1.1.4.1	Gerar Incremento			
14	1.1.4.2	Testar a codificação			
15	1.1.5	Fase de Qualidade			
16	1.1.5.1	Realizar Testes			
17	1.1.6	Reunião de Revisão da Iteração - Apresentação dos modelos e Protótipo	Qui 31/10/19	Qui 31/10/19	Todos
18	1.1.7	Reunião de Retrospectiva da Iteração	Qui 31/10/19	Qui 31/10/19	Grupo E; Grupo F
19	1.2	Iteração 2	Qui 31/10/19	Qui 31/10/19	
20	1.2.1	Fase de Análise de Requisitos			
21	1.2.2	Fase de Qualidade			
22	1.2.3	Fase Projeto de Arquitetura			
23	1.2.4	Fase de Codificação	Qui 07/11/19	Qui 07/11/19	
24	1.2.5	Fase de Qualidade			
25	1.2.6	Reunião de Revisão da Iteração - Apresentação dos modelos e Protótipo	Qui 14/11/19	Qui 14/11/19	Todos
26	1.2.7	Reunião de Retrospectiva da Iteração	Qui 14/11/19	Qui 14/11/19	Grupo E; Grupo F
27	1.3	Iteração 3	Qui 14/11/19	Qui 14/11/19	
28	1.3.1	Fase de Análise de Requisitos	Qui 14/11/19	Qui 14/11/19	
29	1.3.2	Fase de Qualidade			
30	1.3.3	Fase Projeto de Arquitetura	Qui 21/11/19	Qui 21/11/19	
31	1.3.4	Fase de Codificação	Qui 28/11/19	Qui 28/11/19	Todos
32	1.3.5	Reunião de Revisão da Iteração - Apresentação dos modelos e Protótipo	Qui 28/11/19	Qui 28/11/19	
33	1.3.6	Reunião de Retrospectiva da Iteração	Qui 28/11/19	Qui 28/11/19	Grupo E; Grupo F

Figura 5 – Cronograma do *framework Agile O-MaSE*

O *backlog* do produto foi entregue com 22 requisitos, já priorizados pelo dono do produto (*product owner*), representado pela mestrandona. O grupo deveria estimar a complexidade de cada requisito utilizando a técnica de Fibonacci. A cada iteração, uma meta de entrega era passada e os alunos selecionavam os requisitos que seriam desenvolvidos, além de apresentar os artefatos ao final de cada iteração. A figura 6, apresenta o *backlog* do produto, a figura 7, apresenta o *backlog* da iteração e a figura 8, apresenta o plano de *release*.



Backlog do Produto							
Item	Descrição	Temas	Prioridade	Tipo	Estimativa de Complexidade (Fibonacci - 1 a 13)	Iteração	Situação
1	Ao entrar no jogo pela primeira vez, usuário é submetido a questões para definir seu nível e as demais funcionalidades estão indisponíveis.	Nivelamento	10	Requisito	13	1	Pronto
2	A qualquer momento o usuário pode verificar módulos disponíveis para acesso.	Nivelamento, Reforço e Níveis	130	Requisito	5	1	Pronto
3	O usuário visualiza como habilitado apenas módulos disponíveis conforme sua evolução.	Nivelamento, Reforço e Níveis	20	Requisito	1	1	Pronto
4	Ao selecionar o nível disponível na página principal, usuário é direcionado para a primeira questão aleatória do Nível em questão.	Níveis	160	Requisito	1		Pronto
5	Os níveis mais altos têm questões com maior complexidade. As questões disponibilizadas têm níveis de conhecimento de 1 a 5, sendo as questões de nível 1 são mais fáceis do que as questões de nível 5.	Níveis	180	Requisito	2		Pronto

Figura 6 – Backlog do Produto

Na primeira iteração foram desenvolvidos: modelo de metas, modelo de interface com a organização, modelo de papéis, modelo de classes de agentes, de protocolos, além do incremento do software.

BACKLOG DA ITERAÇÃO							
META DA ITERAÇÃO 1	ITEM	HISTÓRIA DE USUÁRIO	TAREFAS DA HISTÓRIA	DURAÇÃO DAS TAREFAS (h)	DISTRIBUIÇÃO DAS TAREFAS	SITUAÇÃO DA TAREFA	
Poder realizar as 10 questões do Nivelamento, responder e ter resposta de questões variadas utilizando texto, imagem, áudio ou vídeo, podendo rever as questões já respondidas, saber das respostas corretas quando estiverem erradas, poder parar e reiniciar o nivelamento e no final receber a orientação de qual nível o usuário deve seguir no jogo.	1	Ao entrar no jogo pela primeira vez, usuário é submetido a questões para definir seu nível e as demais funcionalidades estão indisponíveis.	Criar questões	2		Pronto	
			Criar Login	3		Pronto	
	2		Criar Nivelamento	5		Pronto	
			Implementar	3		Pronto	
			Criar módulos	3		Pronto	
	3		Implementar			Pronto	
	4		Criar Habilitar módulos	1		Pronto	
			Implementar	1		Pronto	
	5		Criar alternativas	1		Pronto	
			Implementar	1		Pronto	
	6		Criar informar opção correta	2		Pronto	
			Implementar	2		Pronto	
	7						
	8						
	9						

Figura 7 – Backlog da Iteração

Plano de Release da metodologia Agile O-MaSE – Grupos E e F
Projeto de Desenvolvimento do Jogo Médico Educacional

Histórico de Revisões

Data	Versão	Descrição	Autor
17/10/2019	1.0	Criação do documento e preenchimento das informações: 1, 2, 3 e 4.	Fabiana Gominho

1. Definição do Critério de Sucesso da Release

Este projeto é orientado por data, e deve ser finalizado no dia 05/12/19. Consequentemente, as entregas das iterações são do tipo *End-Date Driven*, ou seja, orientada para data de término, cujas as entregas devem estar disponíveis antes da data limite.

Os critérios para considerar uma release terminado com sucesso são:

- A entrega de todas as histórias do backlog da iteração;
- A qualidade da entrega: requisitos corretos e sem bugs e
- A apresentação dos modelos e artefatos que a metodologia requer.

O projeto possui um Comitê de Qualidade, composto por: Fabiana Gominho, Vera Werneck e Rosa Costa, que fará a aceitação das entregas de cada iteração.

2. Estimativa dos Itens de Backlog do Produto

A estimativa da complexidade de desenvolvimento deve ser analisada pelo Time de desenvolvimento, item por item do Backlog e deve ser iniciado pelo item de maior prioridade. A técnica utilizada na estimativa dos itens do backlog, deve ser a sequência de número Fibonacci: 1, 2, 3, 5, 8 e 13. Quanto maior for o número, maior será a complexidade de desenvolvimento.

3. Definição do Tamanho e Quantidade de Iterações

O tamanho da iteração está definido limitado a data final do projeto, num time-box de 2 semanas, dando um total de 3 iterações.

4. Estimativa da Velocidade

A estimativa da velocidade será realizada de forma manual, observando a primeira iteração e ajustando as demais iterações.

5. Associação dos Itens de Backlog do Produto as Iterações

Iteração	Backlog do Produto
1	1, 2, 3, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17
2	4, 5, 6, 18, 19, 20, 21, 22

Figura 8 – Plano de Release

O modelo de metas (figura 9) tem como objetivo transformar os requisitos iniciais do sistema em um conjunto de metas estruturadas para o sistema, refinando cada vez mais cada uma das metas à medida que as iterações avançam.



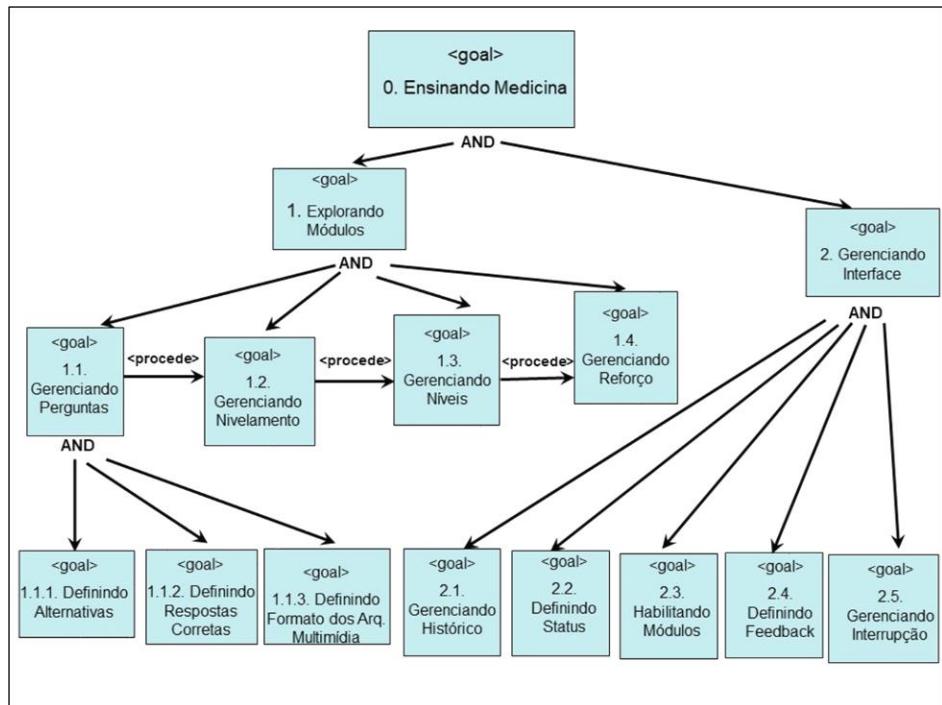


Figura 9 – Modelo de Metas Simplificado

O modelo de papéis (figura 10) identifica os papéis da organização, bem como suas interações entre si e com os atores externos. Foram identificados 4 papéis: avaliador, gerenciador de interface, gerenciador de módulo e gerenciador de perguntas.

O modelo de classe de agentes define o comportamento de um papel em termos de um plano e a capacidade necessária para o agente desempenhar o papel. A figura 11 apresenta o jogador e os agentes, Perguntador, Nivelador e Controlador.

Na segunda iteração, foram atualizados os modelos já existentes, e criado o modelo de políticas, que não havia sido criado na primeira iteração.

Ao final das 3 iterações, foram apresentados além dos modelos finais da metodologia O-MaSE, os *Backlogs* do produto e da Iteração e o Jogo Médico (figuras 12).



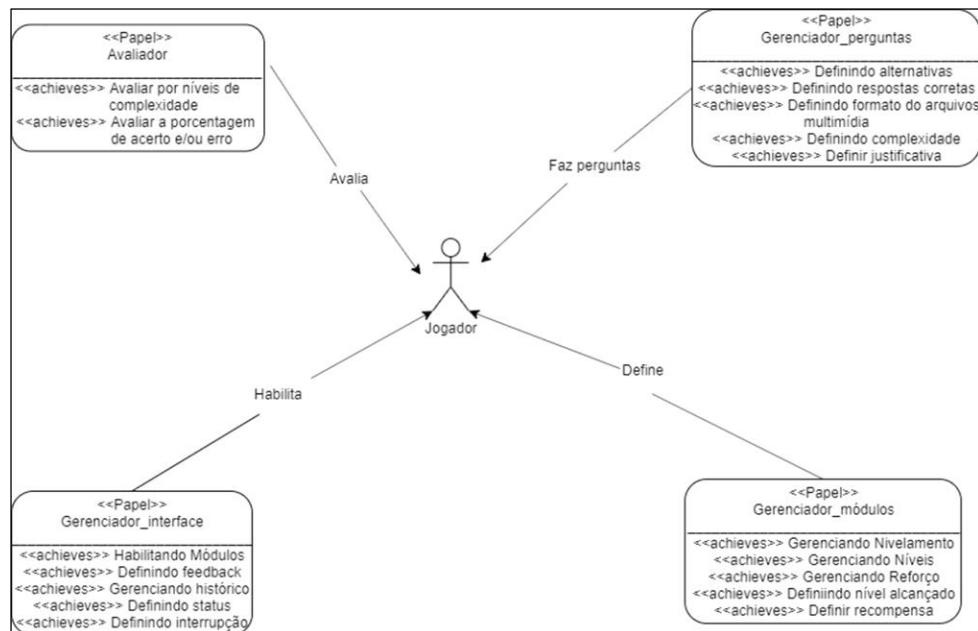


Figura 10 – Modelo de Papéis

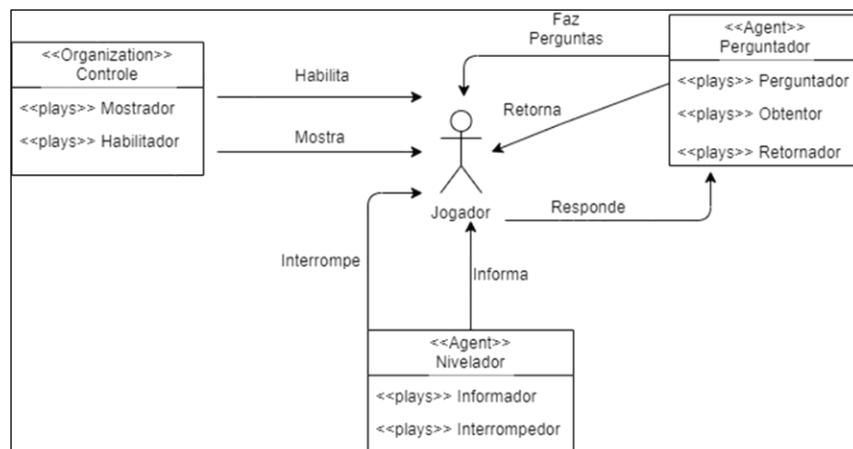


Figura 11 – Modelo de Classe de Agentes

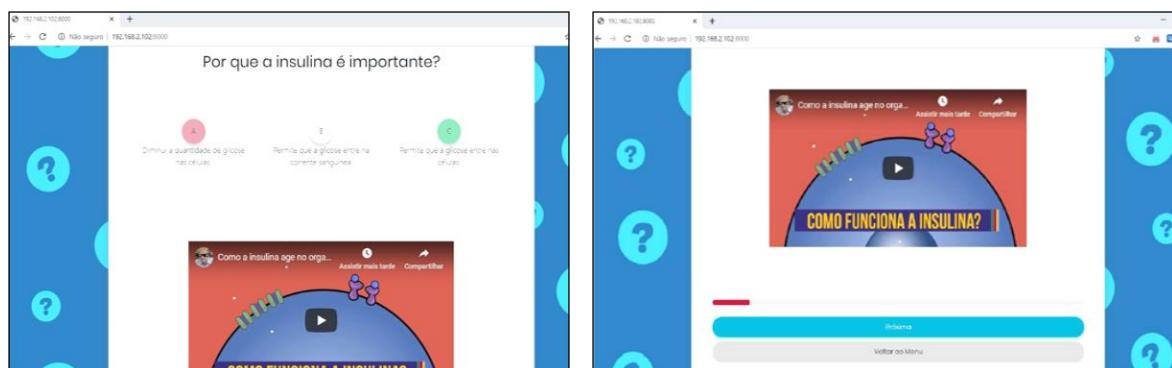


Figura 12 – Telas do Jogo

6. Conclusões e Considerações Finais

Este trabalho teve como principal objetivo descrever uma proposta de adaptação de uma metodologia orientada a agente em um *framework* ágil, fundamentada no Manifesto Ágil e nas boas práticas de *frameworks* ágeis, Scrum e XP.

Realizada a experiência no estudo piloto, junto a outras metodologias ágeis e orientadas a agentes, verificou-se que o processo ágil adaptado foi bem utilizado nas entregas e no entendimento geral da agilidade, porém por estarmos num ambiente acadêmico com o horário restrito da aula e devido à ausência dos alunos em dias que não tinha apresentação da iteração, não foi possível executar o *framework* na íntegra, faltando a realização da prática da reunião de revisão da iteração, que seria realizada após as reuniões de apresentação.

Como todo processo ágil, o empenho da equipe faz a diferença no projeto e a tendência é que o trabalho seja maior no início, com a criação de todos os modelos envolvidos e com o passar das iterações esses produtos precisem apenas de atualizações, até a entrega final dos requisitos.

As características: de controle das iterações ágeis, o modelo adaptativo com eventos de revisão da iteração e retrospectiva da iteração, e a inclusão de uma Fase de qualidade para dar suporte às entregas do projeto, possibilitam a apresentação de um *framework* ágil que possa ser integrado a qualquer método orientado a agentes.

Embora as características: eventos de revisão e retrospectiva da iteração, e a Fase de Qualidade, não estejam contempladas em todos os *frameworks* ágeis existentes, estas são importantes para manter o entrosamento da equipe de projeto com o cliente, a aproximação dos componentes da equipe de projeto, a disciplina para obter um bom incremento do produto, para inspecionar e adaptar o processo e a qualidade das entregas.

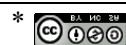
O grupo que realizou o desenvolvimento utilizando a proposta *Agile* O-MaSE, utilizou duas iterações para realizar os 22 requisitos funcionais, porém seria necessário utilizar a terceira iteração para dar mais qualidade ao software, para tratar erros e requisitos não funcionais como: usabilidade e interface externa do jogo.

Como trabalhos futuros, consideramos que a proposta *Agile* O-MaSE poderá ser aplicada num ambiente onde a frequência dos participantes seja mais homogênea, como alunos bolsistas ou estagiários. Além disso, essa proposta será comparada a outros estudos de caso que foram realizados com as metodologias *Agile*-Passi e INGENIAS-Scrum com construção do mesmo jogo.

O estudo piloto permitiu identificar os benefícios da integração desses elementos, que definem a proposta do *framework* *Agile* O-MaSE. Outros estudos poderão ser realizados para aprimorar a proposta, assim como contribuir com mais evidências do funcionamento da proposta genérica do *framework* ágil que possa ser integrado a métodos orientados a agentes.

Referências

AL-AZAWI, Rula; AYESH, Aladdin; OBAIDY, Mohaned Al. Towards agent-based agile approach for game development methodology. In: 2014 World Congress on Computer Applications and Information Systems (WCCAIS). IEEE, 2014. p. 1-6.



* O trabalho Agile O-MaSE: Framework Ágil Orientado a Agentes de Fabiana Ribeiro Ferraz Gominho, Rosa Maria Esteves Moreira da Costa, Vera Maria B. Werneck está licenciado com uma Licença Creative Commons – Atribuição – Não Comercial-Compartilhamento Igual 4.0 Internacional. <http://creativecommons.org/licenses/by-nc-sa/4.0/>

- BABAR, Muhammad et al. The evaluation of agile demand response: An applied methodology. *IEEE Transactions on Smart Grid*, v. 9, n. 6, p. 6118-6127, 2017.
- Beck, Kent; et al. Manifesto for agile software development, 2001. Disponível em: <<http://agilemanifesto.org/>>. Acessado em 27/11/2019.
- CHELLA, Antonio et al. Agile passi: An agile process for designing agents. *International Journal of Computer Systems Science & Engineering*, v. 21, n. 2, p. 133-144, 2006.
- Cossentino, M.; In: IEEE-FIPA. Design Process Documentation Template, jan. 2012. Disponível em: <<http://fipa.org/specs/fipa00097/SC00097B.pdf>>. Acessado em 01/12/2019.
- Deloach, S. A.; Garcia-Ojeda, J. C.; The O-MASE Methodology. In: Cossentino, M. et al. *Handbook on Agent-Oriented Design Processes*. Springer, 2014. p. 253-258.
- FERREIRA, Vitor Manuel Fragoso et al. Developing an educational medical game using AgilePASSI multi-agent methodology. In: 2015 IEEE 28th International Symposium on Computer-Based Medical Systems. IEEE, 2015. p. 298-303.
- GONZÁLEZ-MORENO, Juan C. et al. Ingenias-scrum. In: *Handbook on Agent-Oriented Design Processes*. Springer, Berlin, Heidelberg, 2014. p. 219-251.
- JEFFRIES, Ron; MELNIK, Grigori. Guest Editors' Introduction: TDD--The Art of Fearless Programming. *IEEE Software*, v. 24, n. 3, p. 24-30, 2007.
- NOORI, Fariba; KAZEMIFARD, Mohammad. Simulation of pair programming using multi-agent and MBTI personality model. In: 2015 Sixth International Conference of Cognitive Science (ICCS). IEEE, 2015. p. 29-36.
- NOVO, Fernanda T. et al. Modelando Requisitos de um Jogo Educacional Médico usando a Metodología INGENIAS SCRUM. In: WER-Workshop of Requirements Engineering. 2018.
- STEGHÖFER, Jan-Philipp et al. PosoMAS: an extensible, modular SE process for open self-organising systems. In: International Conference on Principles and Practice of Multi-Agent Systems. Springer, Cham, 2014. p. 1-17.
- Sommerville, I. *Engenharia de Software*. 10. ed. São Paulo: Pearson, 2019.
- TENSO, Tanel et al. Enhancing requirements engineering in agile methodologies by agent-oriented goal models: Two empirical case studies. In: 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, 2017. p. 268-275.
- WANG, Zhe. Teamworking strategies of scrum team: a multi-agent based simulation. In: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence. 2018. p. 404-408.
- YU, Han et al. A multi-agent game for studying human decision-making. In: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems. 2014. p. 1661-1662.

