

A RELAÇÃO ENTRE METODOLOGIAS ÁGEIS E UML NO DESENVOLVIMENTO DE SOFTWARE: UMA REVISÃO SISTEMÁTICA DA LITERATURA

Wellington Kauê Ribeiro de Oliveira¹; Wilson Vendramel²

Resumo

No escopo de metodologias ágeis, a documentação costuma ser um tópico secundário e, por consequência disso, é muitas vezes deixada de lado. Visando mitigar o risco de perda de informação e falhas no compartilhamento de conhecimento, identificou-se a oportunidade de analisar como a documentação visual pode agregar valor ao desenvolvimento ágil. No contexto de artefatos visuais, a UML é vista como uma linguagem de modelagem com bastante potencial para essa finalidade. Este artigo apresenta uma revisão sistemática da literatura que sintetiza e analisa pontos em comum a partir de trabalhos publicados referentes ao uso da UML em conjunto com metodologias ágeis. Como resultado, concluiu-se que uma relação adequada entre o desenvolvimento ágil e diagramas UML pode trazer vantagens para os projetos de software como um todo, especialmente no aspecto de compartilhamento de informações entre os envolvidos no projeto.

Palavras-chave: Metodologias Ágeis; UML; Revisão Sistemática da Literatura.

Abstract

In the scope of agile methodologies, documentation is often a secondary topic and, as a result, is often left out. In order to mitigate the risk of information loss and knowledge sharing failures, the opportunity to analyze how visual documentation can add value to agile development was identified. In the context of visual artifacts, UML is seen as a modeling language with a lot of potential for this purpose. This article presents a systematic review of the literature that synthesizes and analyzes points in common from published works regarding the use of UML in conjunction with agile methodologies. As a result, it was concluded that an adequate relationship between agile development and UML diagrams can bring advantages to software projects as a whole, especially in the aspect of information sharing among those involved in the project.

Keywords: Agile Methodologies; UML; Systematic Literature Review.

1 Introdução

De acordo com Gomes, Willi e Rehem (2014), o entendimento do significado de desenvolvimento ágil de software começa a partir da compreensão do manifesto ágil que propõe alguns valores sobre o que deve ser priorizado ao se desenvolver um produto de software a partir da metodologia ágil, dando origem aos princípios que compõem essa metodologia e permitindo aos desenvolvedores a produção de sistemas de software com requisitos instáveis em períodos relativamente curtos.

¹ Graduado em Análise e Desenvolvimento de Sistemas pela Faculdade de Tecnologia Zona Leste-FATEC; e-mail: wellgn123@gmail.com.

² Doutorando em Tecnologias da Inteligência e Design Digital pela Pontifícia Universidade Católica de São Paulo-PUC/SP, professor da Faculdade de Tecnologia Zona Leste-FATEC, e-mail: wilson.vendramel@fatec.sp.gov.br.

Esses valores são: “Indivíduos e interações mais que processos e ferramentas; Software funcionando mais que documentação abrangente; Colaboração com o cliente mais que negociação de contratos; Responder a mudanças mais que seguir um plano” (MANIFESTO, 2001).

Os métodos ágeis são construídos tendo como base os princípios que complementam os valores no manifesto ágil. Pode-se entender como características principais descritas por esses princípios a entrega contínua de software com valor agregado, as vantagens tiradas das mudanças visando a vantagem competitiva para o cliente, a entrega frequente de software em funcionamento no menor tempo possível, os stakeholders e desenvolvedores trabalhando em conjunto, a motivação dos indivíduos envolvidos no projeto, a transmissão contínua de informação entre a equipe, a manutenção do ritmo de desenvolvimento, a atenção à excelência técnica e ao design a fim de aumentar a eficiência e também a simplicidade no sentido de eliminar o que não é preciso ser feito.

Gomes, Willi e Rehem (2014) enfatizam que o manifesto ágil começa dizendo que “estamos descobrindo melhores maneiras de desenvolver software”, a partir disso pode-se entender que a metodologia ágil está em constante evolução e melhoria, ou seja, a ideia não é propor uma solução definitiva, mas sim um direcionamento do melhor caminho.

O manifesto ágil, seus valores e princípios, trouxeram grandes vantagens e ganhos para as organizações, porém um dos pontos passíveis de melhoria está relacionado à comunicação entre os membros do time de desenvolvimento, e destes com os stakeholders, no sentido da documentação envolvida.

Stettina e Heijstek (2011) e também Stettina, Heijstek e Fægri (2012), demonstram a importância da documentação envolvida nas metodologias ágeis ao realizarem estudos que buscam demonstrar o potencial valor agregado ao se identificar um tipo de documentação adequada ao processo. Os trabalhos em questão também foram utilizados como motivadores para esta pesquisa, a fim de estabelecer a importância da documentação no desenvolvimento ágil de software. Embora os estudos não sejam revisões sistemáticas, a motivação identificada em ambos vem da atitude de discutir e validar a efetividade da documentação em projetos ágeis.

Stettina e Heijstek (2011) elaboraram um estudo empírico buscando elucidar as percepções de um grupo de desenvolvedores de software ágil acerca da documentação em seus projetos. O estudo revelou que aproximadamente metade dos indivíduos acredita que a documentação é um fator importante no processo, mas ressaltam que a documentação disponível em seus projetos é escassa. Mesmo havendo documentos não utilizados diretamente pelo usuário final do produto, uma documentação técnica do projeto de software é de grande

benefício para quem inicia no projeto, bem como para os futuros mantenedores. Ao apresentar as conclusões, o estudo reforça que a documentação na metodologia ágil deve apenas conter informação a ser usada de fato, além da falta de uma documentação apropriada poder acarretar grandes perdas de conhecimento.

Stettina, Heijstek e Fægri (2012) se aprofunda no âmbito da documentação e aponta inicialmente o compartilhamento de informação como a atividade mais afetada em projetos que seguem estritamente o desenvolvimento ágil, reiterando o estudo de Stettina e Heijstek (2011), o qual reforça a falta de uma documentação estruturada como uma das causas para um maior gasto de tempo no processo de manutenção de um produto de software proveniente de um projeto ágil.

Tomando como referência os estudos envolvendo o desenvolvimento ágil e os estudos de Stettina e Heijstek (2011) e Stettina, Heijstek e Fægri (2012), sobre a importância da documentação no desenvolvimento ágil, estabeleceu-se o objetivo de avaliar um tipo de modelagem para produzir artefatos a serem utilizados como documentação para metodologias ágeis e alinhados ao desenvolvimento de modo a agregar valor ao projeto. O valor agregado ao projeto diz respeito à comunicação entre os desenvolvedores, stakeholders e mantenedores do sistema.

A UML é uma linguagem de modelagem para sistemas de software, ela apresenta diversos tipos de diagramas que são usados para descrever diferentes aspectos da modelagem de um sistema, como por exemplo, as classes que compõem o sistema, a interação das classes, o fluxo de controle e de atividades do sistema etc., além de permitir um nível mais alto de abstração no design do software (BOOCH; RUMBAUGH; JACOBSON, 2006).

Uma vez que a UML gera modelos visuais, pode-se atribuir grande potencial de acoplamento às metodologias ágeis, pois artefatos visuais são uma das formas mais eficientes de se identificar os requisitos. Esses modelos auxiliam o desenvolvedor a garantir aos stakeholders o entendimento da solução, de forma a ficarem interessados e engajados, tornando-se elementos essenciais para encontrar falhas nos requisitos (CHEN; BEATTY, 2012). Os autores ainda ressaltam que a visualização do modelo cria uma imagem da solução, desse modo, permite aos stakeholders o entendimento do que será entregue, facilitando a comunicação entre os membros do time e o compartilhamento de informações sobre o projeto.

Para validação da UML como forma de documentação adequada para metodologias ágeis, este trabalho tem o objetivo de realizar uma revisão sistemática da literatura, buscando levantar pontos de concordância e discordância entre estudos que relacionam de forma prática e teórica o uso da UML em conjunto com metodologias ágeis.

O método utilizado para se selecionar e entender os estudos neste artigo segue os critérios de Kitchenham (2004) e Fabbri *et al.* (2017), sendo dividido principalmente em planejamento, condução e documentação. No planejamento é definida a estratégia de busca e a seleção dos estudos iniciais. Na condução, é realizada a recuperação de estudos mais pertinentes, além da leitura, análise e seleção dos textos. Por fim, na documentação é feita a extração e síntese de dados.

Além desta seção de introdução, este trabalho descreve a metodologia de revisão na seção 1, a aplicação da pesquisa na seção 2, a síntese e análise dos resultados na seção 3, e por fim, na seção 4, as conclusões.

2 Metodologia de Revisão

Seguindo o método descrito por Kitchenham (2004) e Fabbri *et al.* (2017), o desenvolvimento da revisão segue basicamente as etapas de planejamento, condução e documentação. Cada uma destas etapas é composta por uma série de atividades para o desenvolvimento do processo.

A etapa de planejamento refere-se à estruturação do plano que direcionará a revisão, identificando sua necessidade, seu contexto, desenvolvimento do protocolo da revisão e modo de validação dos estudos.

A condução é pertinente à recuperação dos estudos primários e definição das formas nas quais a revisão operará. Define-se também nesta etapa, os critérios de inclusão e exclusão dos estudos iniciais. Em seguida, após a seleção dos estudos iniciais, é feita a seleção dos estudos com base na leitura de seu conteúdo, para que se valide a importância, pertinência e qualidade desses estudos por meio de uma categorização.

E por fim, na fase de documentação, é realizada a extração e síntese de dados que constata as informações necessárias para responder à questão de pesquisa.

3 Aplicação da Pesquisa

Esta seção apresenta a aplicação da pesquisa, dividida em: a) questão de pesquisa; b) estratégias de busca; c) restrições de busca; d) resultados das buscas; e) critérios de inclusão e exclusão de estudos; e f) extração e síntese de dados.

a) Questão de pesquisa

Segundo o método de revisão descrito por Fabbri *et al.* (2017), o primeiro passo do planejamento é definir a estratégia dos critérios PICO, estabelecendo assim a questão de pesquisa que é o elemento central de interesse da revisão sistemática, podendo haver uma ou

mais questões de pesquisa. PICO é um acrônimo para representar os elementos fundamentais da questão de pesquisa, sendo eles: a População delimitada ou o Problema definido; o Interesse na população ou a Intervenção no problema; o Controle utilizado na população, incluindo a Comparação de resultados; e o desfecho do processo do inglês *Outcome*. O significado de cada critério da estratégia PICO é apresentado no quadro 1.

Quadro 1 – Significado da estratégia PICO

Sigla	Significado
P	População, Problema
I	Interesse, Intervenção
C	Controle, Comparação
O	Desfecho, Resultado (<i>Outcome</i>)

Fonte: Elaborado pelos autores (2020).

A utilização dessa estratégia tem origem em pesquisa de áreas clínicas como descrito por Takahashi, Saheki e Gardim (2014) e também por Fabbri *et al.* (2017), porém ambos os trabalhos deixam claro que a estratégia pode ser aplicada em outras áreas de pesquisa e descrevem de forma similar o conceito da estratégia PICO, mas Takahashi, Saheki e Gardim (2014) exemplificam a notação dos elementos C e O em conjunto, representando o significado de Contexto, a ser adotado neste trabalho por causa da melhor aderência aos interesses da revisão. O significado de cada critério da estratégia PICo, conforme descrito por Takahashi, Saheki e Gardim (2014) é apresentado no quadro 2.

Quadro 2 – Significado da estratégia PICo

Sigla	Significado
P	População, Problema
I	Interesse, Intervenção
Co	Contexto

Fonte: Elaborado pelos autores (2020).

A questão de pesquisa deve ser formulada pela definição dos critérios da estratégia PICo. Esta definição é apresentada no quadro 3.

Quadro 3 – Atribuição de definição dos elementos PICO

Elemento	Significado
População	Estudos que relacionam metodologias de desenvolvimento ágil com UML.
Interesse	Interação entre os conceitos UML e ágil, de forma prática e teórica
Contexto	Interação entre os conceitos UML e ágil, de forma prática e teórica

Fonte: Elaborado pelos autores (2020).

A partir dos elementos da estratégia PICO, a questão de pesquisa foi definida como: “Como é a relação entre metodologias ágeis e UML, de forma teórica e prática, em trabalhos publicados sobre desenvolvimento ágil de software?”

b) Estratégias de busca

Após a definição da questão de pesquisa, estabeleceu-se a estratégia de busca, possibilitando começar a recuperação de estudos iniciais. No caso deste trabalho, optou-se por utilizar bases digitais para realização das buscas pelo motivo de essas bases possibilitarem a aplicação de critérios de filtragem automatizada. As seguintes bases foram utilizadas:

- sciencedirect.com;
- ieeexplore.ieee.org;
- dl.acm.org;
- link.springer.com.

As strings de busca, conforme descreve o quadro 4, foram aplicadas nas bases digitais de busca, localizando os estudos a partir dos metadados, ou seja, título, resumo, referências e palavras-chave.

Quadro 4 – Relação de strings de busca e seus significados

Identificação	Significado	Significado da String
Str01	(“Agile Methodology” or “Agile Development”) and Documentation	Textos que abordam metodologia ou desenvolvimento ágil relacionados com documentação
Str02	(“Agile Methodology” or “Agile Development”) and UML	Textos que abordam metodologia ou desenvolvimento ágil relacionado com UML

Fonte: Elaborado pelos autores (2020).

A mesma estratégia foi utilizada para todas as bases. Em primeiro lugar aplicou-se as strings de busca, e em seguida definiu-se os parâmetros restritivos de seleção, parâmetros estes detalhados nas restrições de busca.

c) Restrições de busca

Para especificação na recuperação de estudos, as seguintes restrições foram aplicadas como filtro em todas as bases de busca. Este filtro foi aplicado utilizando-se os mecanismos de seleção de cada base.

Restrição 1: o estudo pré-selecionado deve ser um artigo; não pode ser livro, arquivo de áudio, arquivo de vídeo etc.;

Restrição 2: o artigo deve estar relacionado com projeto e desenvolvimento de software; tal seleção deve ser feita utilizando-se do mecanismo de restrição de pesquisa de cada base;

Restrição 3: o artigo deve ter sido publicado após o ano de 2000 por conta de ser mais relevante para o estudo.

d) Resultados das buscas

Ao realizar a recuperação de artigos provenientes da pesquisa com as strings de busca Str01 e Str02, foi observado que, no caso da Str01, foram retornados trabalhos sobre documentação envolvida com metodologias ágeis, entretanto não citando UML necessariamente, evidenciando a ineficácia da Str01 para recuperação de estudos pertinentes. Ao passo que, a string de busca Str02 possibilitou recuperar uma quantidade significativa de estudos condizentes com a temática de metodologias ágeis e UML aqui abordado, conforme mostra o quadro 5. Desta forma foram considerados como estudos inicialmente selecionados apenas os provenientes da Str02, sendo estes resultados contabilizados para a aplicação dos critérios de inclusão e exclusão, a serem detalhados mais adiante.

Quadro 5 – Resultado das buscas nas bases de pesquisa Str02

Base de Busca	String de Busca	Resultados
Science Direct	Str02	58
IEEEExplore	Str02	14
Portal ACM	Str02	14
SpringerLink	Str02	12

Fonte: Elaborado pelos autores (2020).

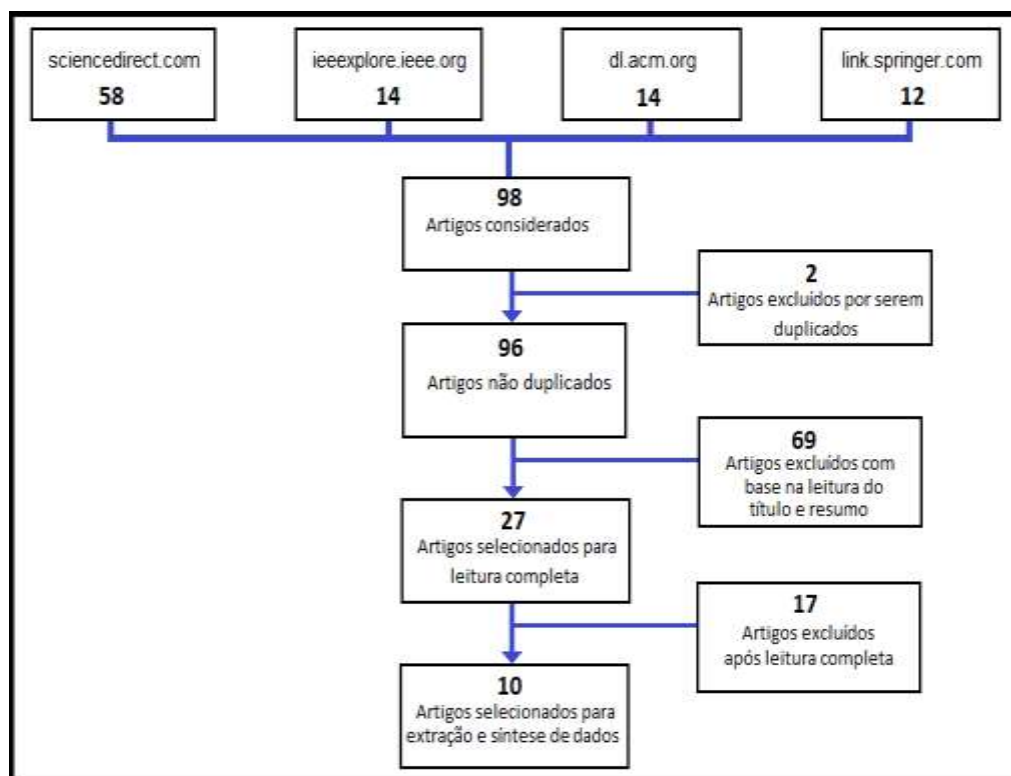
e) Critérios de inclusão e exclusão de estudos

Após a busca de estudo iniciais, foram aplicados critérios para selecionar os retornos mais pertinentes. Esses critérios, listados a seguir, foram definidos com base na metodologia apresentado por Kitchenham (2004) e Fabbri *et al.* (2017).

1. Critério referente aos artigos duplicados: critério utilizado para realizar a remoção de artigos repetidos trazidos das diferentes bases;
2. Critério referente à seleção com base no título e resumo: critério utilizado para validar se o título e o resumo do artigo demonstram que o trabalho apresenta importância e relevância para a pesquisa;
3. Critério referente à seleção com base na leitura do texto: uma vez que a leitura apenas do título não é suficiente para eleger o artigo para a extração de dados, a leitura de seu conteúdo é realizada para assegurar-se da importância do trabalho.

Inicialmente foram selecionados 58 estudos da base Science Direct, 14 da base IEEEExplore, 14 do Portal ACM e 12 da base SpringerLink, totalizando 98 estudos considerados inicialmente, como exibido na figura 1.

Figura 1 – Fluxo de critérios de inclusão e exclusão inicial dos artigos



Fonte: Elaborado pelos autores (2020).

Para a aplicação do 3º critério, foi observada a necessidade de se fazer a separação dos 27 artigos em quatro possíveis grupos de interesse, pois as strings de busca utilizadas trouxeram trabalhos que, embora as contivessem, não possuíam conteúdo relacionando métodos ágeis e

UML de forma relevante. Os artigos de interesse para extração e síntese de dados são os que descrevem de forma teórica, a aplicação de diagramas UML utilizados em metodologias ágeis, assim como os artigos que descrevem a aplicação de ambos os conceitos na prática, sendo assim pertinentes para esta pesquisa. Os possíveis grupos de interesse são:

1. Grupo: artigos sobre modelos ágeis e UML relacionados de forma prática;
2. Grupo: artigos sobre modelos ágeis e UML relacionados de forma teórica;
3. Grupo: artigos sobre a documentação envolvida em metodologias e processos ágeis que não citam a UML;
4. Grupo: artigos mencionando diagramas UML e metodologias ágeis, mas que não fazem relação entre ambos.

Para fins de interesse desta pesquisa, foram considerados apenas o primeiro e segundo grupos, deste modo, os artigos que descrevem a documentação utilizada no processo ágil e não mencionam a UML, e os artigos recuperados que citam a UML e métodos ágeis sem relacioná-los, não são pertinentes para este trabalho.

f) Extração e síntese de dados

A partir dos artigos selecionados pela leitura do texto, foram desconsiderados 17 trabalhos. O próximo passo é a atividade de extração de dados dos artigos mais relevantes, totalizando aqui 10 trabalhos. Essa atividade consiste em extrair as informações mais importantes e pertinentes para o objetivo da pesquisa.

Como parte da atividade de extração de dados, os artigos selecionados foram listados em um formulário de extração de dados para facilitar a leitura dos trabalhos. O formulário teve a principal função de evidenciar o status de análise de cada artigo, bem como listar as informações retiradas de cada um deles. O quadro 6 exhibe a relação dos artigos escolhidos para a extração de dados.

Quadro 6 – Relação de artigos selecionados

BRAUDE, E. Incremental UML for agile development: embedding UML class models in source code. Proceedings of the 3rd International Workshop on Rapid Continuous Software Engineering. IEEE Press, 2017. p. 27-31.

BRAUDE, E.; VAN SCHOONEVELD, J. Incremental UML for agile development with PREXEL. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings. ACM, 2018. p. 310-312.

ELALLAOUI, M.; NAFIL, K.; TOUAHNI, R. Automatic generation of UML sequence diagrams from user stories in Scrum process. Intelligent Systems: Theories and Applications (SITA), 2015 10th International Conference on. IEEE, 2015. p. 1-6.
--

ELALLAOUI, M.; NAFIL, K.; TOUAHNI, R.; MESSOUSSI, R. Automated Model Driven Testing Using AndromDA and UML2 Testing Profile in Scrum Process . Procedia Computer Science, v. 83, p. 221-228, 2016.
HIRANABE, K. Agile Modeling with Mind Map and UML . In: Lecture notes in Computer Science . Springer, Berlin, Heidelberg, 2002. p. 297-309.
LEE, W.; PARK, S.; LEE, K.; LEE, C.; LEE, B.; JUNG, W.; KIM, T.; KIM, H.; WU, C. Agile development of Web application by supporting process execution and extended UML model . Software Engineering Conference, 2005. APSEC'05. 12th Asia- Pacific. IEEE, 2005.
PAIGE, R.; AGARWAL, P.; BROOKE, P. Combining agile practices with UML and EJB: A case study in agile development . International Conference on Extreme Programming and Agile Processes in Software Engineering. Springer, Berlin, Heidelberg, 2003. p. 351-353.
RUMPE, B. Agile modeling with the UML . Radical Innovations of Software and Systems Engineering in the Future. Springer, Berlin, Heidelberg, 2004. p. 297-309.
SANTOS, N.; FERNANDES, J. M.; CARVALHO, M. S.; SILVA, P. V.; FERNANDES, F. A.; REBELO, M. P.; BARBOSA, D.; MAIA, P.; COUTO, M.; MACHADO, R. J. Using Scrum together with UML models: A collaborative University-Industry R&D software project . International Conference on Computational Science and Its Applications. Springer International Publishing, 2016. p. 480-495.
WEI, Q.; DANWEI, G.; YAOHONG, X.; JINGTAO, F.; CHENG, H.; ZHENGANG, J. Research on software development process conjunction of scrum and UML modeling . Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2014 Fourth International Conference on. IEEE, 2014. p. 978-982.

Fonte: Elaborado pelos autores (2020).

Para se fazer a extração de dados foi preciso primeiramente identificar e entender como os artigos descreviam o uso da UML em projetos de metodologia ágil. Para tanto, a partir da leitura integral desses artigos, foram identificadas as formas de uso da UML, sendo considerados pontos comuns entre os trabalhos. Estas formas foram separadas e listadas com a sigla AB, pois se trata de diferentes abordagens de uso da UML:

- AB1: modelo construído a partir da junção de ágil e UML, sendo que os diagramas UML foram incorporados nas etapas do processo ágil;
- AB2: utilização da UML na modelagem de requisitos iniciais do projeto ágil;
- AB3: geração de modelos UML automaticamente a partir do código;
- AB4: uso de arquivos XMI para conversão automática de código para um diagrama UML.

4 Síntese e Análise dos Resultados

Para iniciar a síntese dos dados, uma vez identificados os pontos em comum entre os assuntos abordados pelos artigos, criou-se um quadro de relação entre artigo e assunto, para possibilitar a visualização de quais assuntos são abordados em cada artigo.

Observa-se a partir da leitura do quadro 7 que o mesmo assunto pode ser abordado por mais de um artigo, da mesma forma, um mesmo artigo pode abordar mais de um assunto diferente.

Quadro 7 – Relação dos artigos com os assuntos abordados

	AB1	AB2	AB3	AB4
Braude (2017)			X	
Braude e Schooneveld (2018)			X	X
Elallaoui, Nafil e Touahni (2015)		X	X	X
Elallaoui <i>et al</i> (2016)			X	
Hiranabe (2002)		X	X	
Lee <i>et al.</i> (2005)		X		X
Paige, Agarwal e Brooke (2003)			X	
Rumpe (2004)	X			
Santos <i>et al.</i> (2016)	X	X		
Wei <i>et al.</i> (2014)	X	X		

Fonte: Elaborado pelos autores (2020).

Analisando o quadro 7, é possível verificar que o trabalho de Wei *et al.* (2014), se encaixa bem na abordagem AB1, pois insere a UML em diversas etapas da metodologia Scrum, em especial na fase de planejamento das sprints, utilizando modelos estáticos como diagramas de classes para definição da arquitetura do sistema, além de modelos de casos de uso no *product backlog* devido a sua simplicidade, dinamismo e fácil adaptação, o que o torna interessante para essa fase do Scrum, e também modelos dinâmicos como diagramas de sequência, de atividade, de estado e de colaboração nas fases de implementação. Nesse trabalho observa-se a utilidade de diagramas UML, desde as etapas iniciais até as mais avançadas do projeto, pois conforme o trabalho mostra, tais diagramas se acoplam de forma muito complementar, não somente ao desenvolvimento ágil, mas também aos aspectos de organização do projeto. O método proposto também utiliza a UML para testar as funcionalidades na fase de implementação da sprint. O

processo no trabalho de Wei *et al.* (2014), é chamado de Scrum-uml porque acopla na metodologia Scrum, diagramas provenientes da UML.

Ainda sobre a abordagem AB1, o trabalho de Rumpe (2004) propõe um tipo de desenvolvimento de software baseado em modelagem UML, desse modo, os modelos são adotados como artefatos principais para a documentação, codificação e teste. O estudo estabelece o uso de diagrama de classes para produção de código, e diagrama de objetos em conjunto com diagrama de sequência para produção de testes, além de referenciar diversas vezes o uso de uma versão executável da UML, quando usada dentro de uma abordagem ágil, pode tornar o projeto mais eficiente. Esse artigo entende a UML como um tipo de documentação mais atualizável do que, por exemplo, comentários nos códigos, e por este motivo, mais confiável. A principal contribuição desse trabalho para esta revisão sistemática, vem do fato do mesmo propor uma forma de centralizar a documentação, codificação e teste do software na modelagem UML, o que torna orgânica a integração com uma metodologia ágil, pois torna o processo de documentação, desenvolvimento e teste rápido e menos oneroso.

Por fim sobre a abordagem AB1, pode-se citar o trabalho de Elallaoui *et al.* (2016) o qual foca em testes de desenvolvimento de software, apresentando uma iniciativa para geração automatizada de casos de teste a partir de modelos UML, dentro do processo ágil Scrum, esta iniciativa automatiza duas etapas do processo, primeiro a transformação de modelos de design em modelos de teste, e segundo a geração automática de casos de teste.

Ainda de acordo com a análise do quadro 7, o segundo aspecto mais comum abordado é a modelagem utilizando diagramas UML nas fases iniciais do projeto ágil, ou seja, trabalhos no contexto da abordagem AB2, como é o caso dos estudos de Lee *et al.* (2005), Hiranabe (2002), Elallaoui, Nafil e Touahni (2015), Paige, Agarwal e Brooke (2003), Wei *et al.* (2014) e Santos *et al.* (2016). Apesar de cada um desses estudos fazer utilização dessa abordagem de modo semelhante, Lee *et al.* (2005) se destaca por detalhar muito bem o uso dos diagramas UML na fase de planejamento do projeto. O trabalho apresenta o conceito ágil chamado AWE do inglês, *Agile Web Engineering*, que é adaptado para melhor se encaixar nas necessidades. A fase de planejamento é vista como sendo de modelagem navegacional, estrutural e arquitetural. Segundo o trabalho, o modelo navegacional utiliza diagrama de estados, o modelo estrutural utiliza diagrama de classes e o modelo arquitetural utiliza diagrama de componentes, deixando claro a efetividade da integração da UML no projeto ágil.

Seguindo a análise do quadro 7, constatou-se que a abordagem AB3 tem sido a mais utilizada, estando também em consonância com a abordagem AB1, pois de acordo com os trabalhos de Braude (2017) e Braude e Schooneveld (2018), os diagramas UML são gerados a

partir do código das funções de maior importância do sistema. Braude (2017) utiliza um método de marcação de caracteres, tendo como ponto positivo não apresentar grande complexidade gráfica nos diagramas UML, possibilitando rápido entendimento dos diagramas. Segundo esse método, o diagrama UML é gerado a partir das classes e funções do próprio código. Os nomes das classes no código passam a ser os nomes das classes definidos no diagrama, da mesma forma as operações de cada classe são listadas. O trabalho segue o preceito de criar apenas os fragmentos UML necessários para as classes mais importantes do sistema. Para deixar claro, os modelos descritos no trabalho são apresentados de forma bem resumida usando a UML para representar principalmente as relações das classes mais significativas do sistema. O trabalho tem esse ponto a ser considerado como negativo, pois no intuito de retratar o método implicitamente, passa a requerer do leitor um conhecimento bastante específico. Os modelos se encaixam na notação necessária das regras de modelagem UML, pois seguem o padrão de diagrama de classes mostrando o nome das classes e suas operações.

Em similaridade com a abordagem AB3, Braude e Schooneveld (2018), apresenta um modo de geração de modelos UML criados simultaneamente com a programação. Para isso é utilizado uma ferramenta cujo propósito é implementar uma codificação específica para gerar automaticamente diagramas de classe UML nos pontos de especificação utilizados nas classes, permitindo assim gerar estruturas de classes e métodos, incluindo as relações entre classes, e salvar esses modelos em um arquivo à parte. Esse trabalho apresenta uma forma de integrar a UML com qualquer metodologia ágil, pois é possível gerar os modelos enquanto a codificação é feita, tornando o processo de criação de modelos UML dinâmico e rápido. É possível considerar o estudo de Braude e Schooneveld (2018) como um dos mais importantes nesta pesquisa, embora seja sucinto, apresenta uma forma harmônica, direta e simples, de integrar a modelagem UML em qualquer projeto ágil.

Ainda no contexto da abordagem AB3, Paige, Agarwal e Brooke (2003), constroem um diagrama de classes, em paralelo ao Enterprise JavaBeans (EJB), componente JEE cujo papel é fornecer um desenvolvimento rápido e simplificado de aplicações Java, os quais são utilizados para descrever o projeto posteriormente. Diagramas de classe são usados para modelagem de especificação em vez de modelagem conceitual. As classes dos diagramas de classes são identificadas como seções e entidades do EJB. Diagramas de colaboração são utilizados para entender a lógica de negócios complexa. Segundo o trabalho, os diagramas UML também podem ser utilizados para ajudar a explicar o funcionamento do sistema para o cliente, uma vez que o EJB apresenta grande complexidade. Após desenvolver os argumentos da metodologia proposta em seu trabalho, os autores demonstram a efetividade dessa metodologia por meio do

estudo de caso realizado, expondo que o estudo se mostrou significativo, tomando como base, prazos e adequação aos requisitos-chave estabelecidos no projeto.

No que tange à abordagem AB3, Hiranabe (2002), utiliza um método de criação de modelos UML a partir de anotações de alto nível representadas por meio de mapas mentais (*mind maps*). Os mapas mentais são utilizados para coletar dados com o cliente/usuário e podem ser aplicados de forma interativa. Fazer o mapa mental consiste em estabelecer um termo ou figura central de interesse e em seguida alocar a este termo ou figura, outros termos associativos representam as formas de alcançar o interesse e objetivo do cliente. Nesse trabalho, é utilizado um recurso de conversão automática de mapas mentais em diagramas UML. O método adotado se alinha ao Manifesto (2001), o qual afirma que indivíduos e interações entre eles são mais importantes que processos e ferramentas, pelo fato de a comunicação ser constante com o cliente, através da interação na elaboração do mapa mental, além de possibilitar alto dinamismo referente às expectativas do cliente, propiciando fácil troca de informações entre os envolvidos.

Por fim, sobre a abordagem AB4, é possível integrar a UML ao desenvolvimento ágil, como mostra principalmente Elallaoui, Nafil e Touahni (2015), o qual em seu trabalho automatiza a geração de diagramas UML a partir do código e de diversas formas, como por exemplo, utilizando marcações de arquivos XMI, sendo vista como uma forma bastante produtiva de automatização, embora exija um conhecimento específico sobre esse tipo de arquivo.

5 Conclusão

O desenvolvimento ágil costuma ter uma documentação restrita, abrindo oportunidades para se identificar artefatos visuais de documentação que visam agregar valor ao processo. Os estudos relacionam o uso da UML no contexto das metodologias ágeis de desenvolvimento de software.

Este trabalho visou responder a seguinte questão de pesquisa: “Como é a relação entre metodologias ágeis e UML, de forma teórica e prática, em trabalhos publicados sobre desenvolvimento ágil de software?”. Por meio da síntese e análise dos resultados da extração de dados dos estudos relacionados, tornou-se evidente que uma relação adequada entre a modelagem UML e as metodologias ágeis traz vantagens consideráveis aos projetos de desenvolvimento ágil. Enquanto o desenvolvimento ágil facilita e acelera o desenvolvimento de aplicações de software, os artefatos da UML podem agregar grande valor e qualidade ao processo de desenvolvimento, principalmente na comunicação entre os desenvolvedores, stakeholders e mantenedores do sistema.

Em suma, as vantagens de se relacionar adequadamente a UML com as metodologias ágeis vão desde a facilitação do trânsito das informações até a melhoria do compartilhamento do conhecimento do projeto entre todos os stakeholders. Além disto, esta relação propicia a rastreabilidade dos artefatos gerados ao longo do processo de desenvolvimento, reduzindo o retrabalho ao fazer manutenção no produto de software.

Para trabalhos futuros sugere-se a aplicação de uma pesquisa com profissionais que atuam em projetos ágeis de desenvolvimento de software a fim de se avaliar a aplicabilidade e o grau de relevância da modelagem UML na prática do dia a dia do desenvolvimento de software ágil.

Referências

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. Elsevier Brasil, 2006.

BRAUDE, E. **Incremental UML for agile development: embedding UML class models in source code**. Proceedings of the 3rd International Workshop on Rapid Continuous Software Engineering. IEEE Press, 2017. p. 27-31.

BRAUDE, E.; VAN SCHOONEVELD, J. **Incremental UML for agile development with PREXEL**. In: Proceedings of the 40th International Conference on Software Engineering: Companion Proceedings. ACM, 2018. p. 310-312.

CHEN, A.; BEATTY, J. **Visual models for software requirements**. Pearson Education, 2012.

ELALLAOUI, M.; NAFIL, K.; TOUAHNI, R. **Automatic generation of UML sequence diagrams from user stories in Scrum process**. Intelligent Systems: Theories and Applications (SITA), 2015 10th International Conference on. IEEE, 2015. p. 1-6.

ELALLAOUI, M.; NAFIL, K.; TOUAHNI, R.; MESSOUSSI, R. **automated model driven testing using AndroMDA and UML2 testing profile in Scrum process**. Procedia Computer Science, v. 83, p. 221-228, 2016.

FABBRI, S. C. P. F. *et al.* **Revisão sistemática da literatura em engenharia de software: teoria e prática**. Elsevier Brasil, 2017.

GOMES, A.; WILLI, R.; REHEM, S. O Manifesto Ágil. In: PRIKLADNICKI, Rafael; WILL, Renato; MILANI, Fabiano. **Métodos ágeis para desenvolvimento de software**. Bookman Editora, 2014. p. 4-15.

HIRANABE, K. **Agile modeling with mind map and UML**. In: **Lecture notes in computer science**. Springer, Berlin, Heidelberg, 2002. p. 297-309.

KITCHENHAM, B. **Procedures for performing systematic reviews**. Keele, UK, Keele University, v. 33, n. 2004, p. 1-26, 2004.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering**. Technical report, Ver. 2.3 EBSE Technical Report. EBSE. sn, 2007.

LEE, W.; PARK, S.; LEE, K.; LEE, C.; LEE, B.; JUNG, W.; KIM, T.; KIM, H.; WU, C. **Agile development of web application by supporting process execution and extended UML model**. Software Engineering Conference, 2005. APSEC'05. 12th Asia- Pacific. IEEE, 2005.

MANIFESTO para desenvolvimento ágil de software. 2001. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: maio. 2019.

PAIGE, R.; AGARWAL, P.; BROOKE, P. **Combining agile practices with UML and EJB: A case study in agile development**. International Conference on Extreme Programming and Agile Processes in Software Engineering. Springer, Berlin, Heidelberg, 2003. p. 351-353.

RUMPE, B. **Agile modeling with the UML**. Radical Innovations of Software and Systems Engineering in the Future. Springer, Berlin, Heidelberg, 2004. p. 297-309.

SANTOS, N.; FERNANDES, J. M.; CARVALHO, M. S.; SILVA, P. V.; FERNANDES, F. A.; REBELO, M. P.; BARBOSA, D.; MAIA, P.; COUTO, M.; MACHADO, R. J. **Using Scrum together with UML models: A collaborative University-Industry R&D software project**. International Conference on Computational Science and Its Applications. Springer International Publishing, 2016. p. 480-495.

STETTINA, C. J.; HEIJSTEK, W. Necessary and neglected? An empirical study of internal documentation in agile software development teams. In: **Proceedings of the 29th ACM International Conference on Design of Communication**. 2011. p. 159-166.

STETTINA, C. J.; HEIJSTEK, W.; FÆGRI, T. E. **Documentation work in agile teams: the role of documentation formalism in achieving a sustainable practice**. Agile Conference (AGILE), 2012. IEEE, 2012. p. 31-40.

TAKAHASHI, J.; SAHEKI, Y.; GARDIM, S. O que é PICO e PICo? **Biblioteca “Wanda Aguiar Horta” - Escola de Enfermagem - Universidade de São Paulo**. 2014. Disponível em: <<https://pt.slideshare.net/bibliotecae/o-que-pico-e-pico>> Acesso em: maio. 2019.

WEI, Q.; DANWEI, G.; YAOHONG, X.; JINGTAO, F.; CHENG, H.; ZHENGANG, J. **Research on software development process conjunction of scrum and UML modeling**. Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2014 Fourth International Conference on. IEEE, 2014. p. 978-982.

WEI, Q.; DANWEI, G.; YAOHONG, X.; JINGTAO, F.; CHENG, H.; ZHENGANG, J. **Research on software development process conjunction of scrum and UML modeling**. Instrumentation and Measurement, Computer, Communication and Control (IMCCC), 2014 Fourth International Conference on. IEEE, 2014. p. 978-982.