

Aplicação da UML no contexto das metodologias ágeis

Alan Martins Xavier¹, Fábio Rodrigues Martins¹, Ricardo Bitencourt Pimentel¹,
Denis Rocha de Carvalho¹

Instituto Federal Minas Gerais – Campus São João Evangelista (IFMG-SJE)
Av. Primeiro de Junho 1043 – 39.705-000 – São João Evangelista – MG – Brasil

alanmartinsxavier@gmail.com,
{fabio.martins,ricardo.pimentel,denis.carvalho}@ifmg.edu.br

Resumo. *A UML é um modelo de documentação bastante estudado nos cursos superiores da área de computação, mas as metodologias ágeis ocupam grande parte do mercado atual e elas em seu manifesto pregam por simplicidade, rapidez, entre outras características que vão contra processos minuciosos de documentação. Esse trabalho apresenta os resultados obtidos ao se aplicar um questionário em várias empresas de desenvolvimento de software no estado de Minas Gerais com intenção de descobrir a aplicação da UML no contexto dos processos ágeis de desenvolvimento de software. Os resultados estatísticos obtidos apontam a necessidade de revisão em vários paradigmas da Engenharia de Software.*

1. Introdução

Há quem diga que não consegue viver sem o uso de tecnologia. Nosso cotidiano está repleto de tecnologias e em sua maioria tem como espinha dorsal um software. O software é definido como um conjunto de instruções que ao serem executadas fornecem um retorno esperado [Pressman 2016].

Desenvolver um software não é uma tarefa trivial [Carvalho e Braga 2015] e segundo Sommerville (2011) existem predominantemente duas classificações para processos de desenvolvimento de software: i) dirigidos a planos ou tradicionais ou ii) ágeis.

Ambos os processos podem ser apoiados pela *Unified Modeling Language* (UML), pois trata-se de uma ferramenta independente e é aderente aos processos de desenvolvimento de uma forma geral [Vicari 2012].

Os indicativos do mercado, apontam para uma crescente aderência aos processos de desenvolvimento ágeis. Numa pesquisa realizada em 2011, no sul do Brasil, apontou que 75% das empresas usam processos ágeis [Lima 2011] e a UML não foi citada como uma ferramenta de apoio à modelagem do projeto.

A utilização da UML nos projetos ágeis é uma questão importante para o futuro da Engenharia de Software, pois pode tornar-se um indicador da mudança da experiência do profissional de desenvolvimento de software. Buscando maior agilidade, qualidade e mudando a forma de modelar software sem perder a qualidade e a confiança na manutenção dos requisitos do software.

O mercado mundial de software movimenta mais de US\$ 840 Bilhões, a movimentação nacional ultrapassa US\$ 21 Bilhões. Assim, o Brasil ocupa a 21ª posição

no *ranking* mundial de produção de software. Com uma movimentação tão expressiva, a qualidade do software é um fator primordial [Carvalho e Braga 2015].

É um grande equívoco, associar metodologias ágeis com falta de documentação e falta de qualidade. Pois, as metodologias ágeis possuem técnicas avançadas para garantir a qualidade do produto de software a ser gerado [Sommerville 2011].

É notório que a UML tem maior utilização com os processos tradicionais [Vicari 2012] e que a utilização da UML é uma boa prática na Engenharia de Software [Sommerville, 2011]. Também é notório a grande adoção aos processos ágeis no cenário de desenvolvimento de software e que essa metodologia possui técnicas que apoiam o desenvolvimento de software com qualidade. O que resta saber é o grau de utilização da UML no contexto metodologias ágeis. A resposta desta pergunta, abre uma nova realidade na Engenharia de Software. Com uma mudança da experiência do desenvolvimento de software e nos processos adjacentes, tanto práticos quanto educacionais. Petre (2013) realizou um estudo que apontou para o declínio da UML.

Este trabalho visa fornecer uma visão atualizada do grau de utilização da UML na indústria de software no contexto dos métodos ágeis. Ele está organizado da seguinte forma. A seção 2 apresenta a revisão da literatura. A seção 3 apresenta a metodologia aplicada no estudo. A seção 4 apresenta o resultado da pesquisa e sua discussão. Por fim a seção 5 apresenta a conclusão do trabalho e a seção 6 as referências bibliográficas utilizadas na pesquisa.

2. Revisão da Literatura

2.1. Processos de Software

Um processo de software segundo o *Institute of Electrical and Electronics Engineers* (IEEE), é uma “sequência de passos executados com um determinado objetivo”. Já o *Capability Maturity Model Integration* (CMMI), define processo como “um conjunto de ações e atividades inter-relacionadas realizadas para obter um conjunto especificado de produtos, resultados ou serviços” [Filho 2009]. Mesmo existindo várias visões, sobre a definição, observa-se o foco comum na sistematização operacional para atingir um objetivo.

Para facilitar o entendimento dos processos de software, existe uma classificação: i) dirigidos a planos ou tradicionais e ii) ágeis. Os processos dirigidos a planos ou tradicionais são processos que as atividades são planejadas antecipadamente, são mais burocráticos. Já os processos ágeis o planejamento é feito gradativamente, possui menos burocracia em relação aos processos tradicionais [Sommerville 2011].

2.1.1. Processos Tradicionais

Os processos contêm fases como especificação, codificação, testes e manutenção. Seguindo esse modelo, foi estabelecido o conceito de modelo ciclo de vida. Que sintetiza os conceitos apresentados de processo de software, ou seja, um sinônimo para processo de software [Filho 2009].

Segundo Sommerville (2011) os principais modelos de ciclos de vida ou modelos de processos de software são:

i) Cascata: O modelo Cascata tem como principal característica do ciclo de vida é a execução sequencial de suas fases e entrega final do produto, ou seja, não existem entregas preliminares;

ii) Incremental: O modelo Incremental muda a experiência de desenvolvimento, pois antes com o Cascata, a entrega era única. Já no desenvolvimento incremental, as entregas são parciais. A principal característica deste modelo é a maior aproximação com o cliente e como consequência, um rápido *feedback*.

A evolução dos processos tradicionais resultou na criação do *Rational Unified Process* (RUP), que nasceu dos trabalhos da criação da UML e do *Unified Software Development Process* em 2003 [IBM 2007]. É tido pela comunidade, como um processo híbrido, pois agrega características de vários modelos de ciclo de vida (Cascata, Incremental, etc) e é um dos principais processos de software do mercado [Sommerville 2011].

2.1.2. Processos Ágeis

A realidade no mercado de software propôs inúmeras alterações tanto na forma de utilização, ou na forma de desenvolver ou adquirir um software. O maior clamor da relação Consumidor de software e Fabricante de software, é a resposta a mudanças [Sommerville 2011].

O impacto deste clamor atinge vários fatores: novas oportunidades de negócio, adequação às regulamentações econômicas e legislativas, requisitos instáveis, etc. Desta forma, foi necessário adotar uma nova estratégia de desenvolver software, assim, diante deste cenário, surgiram as metodologias ágeis.

A grande insatisfação da comunidade de desenvolvimento, com esse tipo de abordagem, resultou na década de 1990, na criação dos métodos ágeis. Tem como base a abordagem incremental (tanto no ciclo de vida, quanto nas entregas) e tem total aderência ao tipo do software onde seus requisitos são instáveis. O objetivo principal é entregar o software em funcionamento o mais rápido possível ao cliente. Também reduzir a burocracia do processo de desenvolvimento, com requisitos de maior valor ao cliente e documentação aderente ao requisito. Existem vários métodos ágeis, os mais conhecidos são o *Extreme Programming* (XP), Scrum, Kanban, Crystal, dentre outros [Sommerville 2011].

2.2. UML

A UML é reconhecida como a linguagem padrão para especificar, visualizar, documentar e construir artefatos de um sistema e pode ser utilizada com qualquer processo de desenvolvimento e com diferentes tecnologias de implementação [Furlan 1998].

A UML estabelece um padrão de modelagem de projetos de sistemas, incluindo seus aspectos conceituais, além das classes escritas em determinada linguagem de programação, processos de banco de dados e componentes de software reutilizáveis.

No final da década de 1980, a quantidade de métodos orientados a objetos saltou de 10 para 50 em aproximadamente 5 anos. Houve uma grande dificuldade para encontrar um método que atendesse às expectativas. Os principais métodos que se destacavam na época eram: i) Booch; ii) OSSE (*Object-Oriented Software Engineering*); e iii) OMT (*Object Modeling Technique*). Respectivamente, criados por

Grady Booch, Ivar Jacobson e James Rumbaugh. Os futuros criadores da UML [Booch 2012].

Oficialmente, a UML foi iniciada em 1994, quando Rumbaugh veio trabalhar na Rational juntamente com Booch. O foco neste momento era a unificação dos seus métodos. A UML apresenta diagramas que, em conjunto, modelam todo o sistema [Furlan 1998]. A seguir seus principais diagramas: Diagrama de Classe; Diagrama de Caso de Uso; Diagramas de Sequência e Diagrama de Atividade [Booch 2012].

A UML pode ser usada para modelar várias fases de um sistema, desde os primeiros contatos até a geração do código. É aplicada em qualquer tipo de sistemas em termos de diagramas de orientação a objeto.

3. Metodologia

No presente trabalho foi adotado o método de pesquisa de levantamento. Isso dar-se-á pelos dados já existentes, que foram buscados diretamente no ambiente empresarial, através de observações, aplicação de questionários e medições [Wazlawick 2014]. Produzindo assim, resultados estatísticos quantitativos, que consolidam os dados da utilização da UML em processos de desenvolvimento ágeis.

A metodologia empregada, em concordância com o plano do projeto, consistiu nas seguintes etapas: 1- Revisão teórica: levantamento do estado da arte referente ao projeto; 2- Elaboração do questionário: elaboração de um questionário capaz de extrair dados relevantes para a pesquisa; 3- Aplicação do questionário: realização de uma pesquisa das empresas de desenvolvimento de software para fins de criar uma população, assim, posteriormente, enviar para cada empresa o questionário; 4- Análise dos dados: realização da análise quantitativa dos dados e sua consolidação para posterior publicação.

A etapa da revisão teórica produziu o conhecimento necessário para entender os processos de desenvolvimento tradicionais e ágeis, além da UML. Parte do resultado desta etapa está disponível na seção 2 deste trabalho.

A etapa da elaboração do questionário ora realizada, culminou na produção de um questionário capaz de extrair os dados do ambiente. O questionário abordou as seguintes visões: i) Identificação da empresa: com o objetivo de conhecer a empresa participante; ii) Seção técnica: com o objetivo de conhecer dados referentes ao processo de desenvolvimento e sua documentação; iii) Uso da UML: com o objetivo de conhecer a utilização da UML no contexto da empresa participante e iv) Contra utilização da UML e avaliação aberta da pesquisa: com o objetivo de entender os motivos de uma não utilização da UML e também abrir um espaço para um depoimento do participante.

A próxima etapa, a aplicação do questionário produziu 21 (vinte e uma) respostas, criando assim a população da pesquisa. O questionário ficou disponível 90 (noventa) dias em um formulário eletrônico disponível na Internet. A etapa da análise dos dados será apresenta na próxima seção.

4. Resultados e discussões

Como apresentado na seção 3, o questionário foi elaborado em 4 (quatro) visões: i) Identificação da empresa; ii) Seção técnica; iii) Uso da UML e iv) Contra utilização da

UML e avaliação aberta da pesquisa. Esta seção será apresentada de acordo com as visões, proporcionando um melhor entendimento da pesquisa.

4.1. Identificação da empresa

A amostragem da pesquisa foi diversificada, pois foi identificada uma grande heterogeneidade de áreas de atuação: Empresas desenvolvedoras de software financeiro (14,3%); Empresas desenvolvedoras de software empresarial (14,3%); Empresas desenvolvedoras de software para gestão de saúde (9,5%); Empresas desenvolvedoras de software para gestão educacional (4,8%); Empresas desenvolvedoras de software customizado (38,1%); Empresas desenvolvedoras de software embarcado (4,8%); Empresas desenvolvedoras de software entretenimento (4,8%); Banco Digital (4,8%) e Empresa de Comunicação: Jornais e Televisão (4,8%).

Dessas empresas participantes, com relação ao seu porte, a pesquisa contou com a participação de empresas de porte pequeno (19%), médio (38,1%) e grande (42,9%). Esta informação indica que o tema da pesquisa é de interesse independentemente do porte, porém, é possível identificar que as empresas de grande porte, possuem maior interesse.

Como apresentado acima, apesar da grande diversificação das áreas de atuação das empresas participantes, seus profissionais são em sua totalidade ligados ao desenvolvimento de software, por exemplo: Analistas de Sistemas; Scrum Master; *Agile Coach*; Engenheiro de Software, dentre outros.

4.2. Seção técnica

Nesta visão foram elaboradas 4 (quatro) perguntas: i) Existe uma grande tendência de uso de Metodologia ágil, no mercado brasileiro atual. A empresa ou organização a qual você está ligada trabalha com essa metodologia?; ii) Qual?; iii) No processo de desenvolvimento adotado por sua empresa é realizado algum tipo de documentação das funcionalidades? e iv) No processo de documentação é utilizado a UML?.

Com este escopo de perguntas, é possível entender o contexto técnico da empresa, desde a utilização de alguma metodologia ágil, ao uso de alguma forma de documentação das funcionalidades e se a UML é utilizada nesse contexto. Logo, após a aplicação do questionário, 100% das empresas participantes usam alguma metodologia ágil. A resposta ao questionamento sobre qual metodologia ágil utilizada, foi diversificada: Scrum + Kanban (47,6%); Scrum + Kanban + DevOps (28,6%); Scrum (19%); Scrum + Kanban + DevOps + Lean (4,8%). Esse resultado mostra que as metodologias ágeis possuem qualidade que podem ser aliadas para alcançar maiores resultados, ou seja, existe um enorme esforço de adaptabilidade para suprir todas as demandas das empresas.

A documentação das funcionalidades apresentou um resultado aderente ao trecho do manifesto ágil, “Software funcionando, do que documentação abrangente” [Sommerville 2011], onde: Possuem pouca documentação (71,4%); Possuem alguns desenhos (19%) e Possuem documentação completa (9,5%). Nessa questão, é claro que as empresas “bem ou mal”, fazem algum tipo de documentação. Logo, é possível verificar que o entendimento do trecho do manifesto ágil apresentado acima foi bem entendido pelas empresas, pois é valorizado mais o software funcionando do que documentação abrangente. Essa valorização, não exclui a existência de uma documentação.

Por fim, a última questão desta visão busca entender se a UML era utilizada, assim, 71,4% das empresas informaram que não usam a UML. Apenas 28,6% das empresas usam a UML em algum momento.

4.3. Uso da UML

Na visão do uso da UML foram elaboradas 2 (duas) perguntas. Buscou-se entender quais os diagramas são utilizados e se o conhecimento sobre UML é levado em consideração na contratação do profissional.

Inicialmente é importante lembrar da última questão da visão anterior, vista na subseção 4.2., onde foi questionado se a UML era utilizada. De acordo com o resultado da pesquisa 71,4% dos participantes não usam a UML e apenas 28,6% usam a UML. Neste cenário, de uma amostragem de 21 (Vinte e uma) respostas, essas caem para 6 (seis) respostas. Desta forma, a análise estatística é delimitada à sub amostragem.

A primeira pergunta buscou entender quais são os diagramas mais utilizados da UML. Foram citados na pesquisa: Caso de uso (83,3%); Classe (100%); Objetos (50%); Colaboração (16,7%); Sequência (50%); Atividades (66,7%); Componentes (16,7%) e Pacotes (33,3%). Os diagramas Estados e Depuração não foram citados. Esse resultado apresenta informações relevantes acerca da utilização dos diagramas, com o destaque para os seguintes diagramas: Diagrama de Classes é utilizado por 100% dos participantes; Diagrama de Caso de uso é utilizado por 83,3% dos participantes; Diagrama de Atividades é utilizado por 66,7% dos participantes.

Ainda tendo como base amostral acima, os participantes responderam a segunda pergunta. Que buscou entender se os conhecimentos acerca da UML eram necessários para contratação do profissional. Assim, 50% responderam que sim e os outros 50% responderam que não. Não houve relação da quantidade de diagramas citados na pergunta anterior para entender se os conhecimentos sobre UML são ou não necessários para a contratação, pois em determinadas respostas o participante citou 4 (quatro) diagramas e na próxima pergunta informou que os conhecimentos sobre UML não são necessários e em outra resposta o participante citou 3 (três) diagramas e na próxima pergunta informou que os conhecimentos sobre UML são necessários. O que faz entender que a necessidade do conhecimento sobre UML é algo atrelado a cultura da empresa.

4.4. Contra utilização da UML e avaliação aberta da pesquisa

Na visão contra utilização e avaliação aberta da pesquisa, foram elaboradas 3 (três) perguntas. Como na visão 4.2 apontou que 71,4% dos participantes não usam a UML, é fundamental para a pesquisa entender os motivos para a contra utilização. Desta forma, foi elaborada a questão que visa entender o motivo da não utilização. A próxima questão busca entender qual estratégia é usada para suprir a ausência da UML e por fim, foi elaborada uma questão para o participante avaliar a pesquisa e realizar alguma contribuição extra.

A primeira questão reportou uma grande variedade de respostas, por exemplo: “A complexidade da utilização dos diagramas assim como a sua quantidade”; “Pela falta do conhecimento sobre a mesma”; “A dificuldade de entendimento”; “A pouca praticidade”; “Documentação de qualidade requer tempo e capacidade do profissional, no dia a dia do desenvolvimento a cobrança e prazos são extremamente apertados por conta disso a documentação é negligenciada”; “O valor gerado pelo UML não

compensa tanto em relação ao esforço pra gerá-lo”; “Não sentimos necessidade, outros meios tem se mostrado suficientes”; entre outros. O destaque destes motivos são: i) “A complexidade da utilização e quantidade de diagramas”, foi citado 23,8% das respostas; ii) “Pela falta de conhecimento sobre a mesma”, foi citado 14,3% das respostas. Esse resultado aponta para duas hipóteses: i) A UML está ultrapassada, o mundo está dinâmico e complexidade em demasia diminui o desempenho do desenvolvimento e ii) O ensino da UML não está atendendo às reais necessidades dos profissionais.

A segunda questão, buscou captar as alternativas para suprir a necessidade de documentação. Assim como a questão anterior, as respostas foram variadas, por exemplo: Histórias de usuário; Diagramas lógicos; *Behavior Driven Development* (BDD); Mapa de jornada de usuário; *User eXperience* (UX); Relatório próprio; Narrativa de negócio + técnica, *Business Process Model and Notation* (BPMN).

Por fim, a última questão, buscou um cenário onde os participantes se expressassem abertamente sobre a pesquisa, onde puderam se expressar abertamente sobre a pesquisa. Seguem algumas participações: “Eu pessoalmente gosto muito da UML, mas vejo muitos problemas nos seus diagramas e eles acabam causando em muitas situações um excesso de trabalho inútil. Nos desenvolvemos sistemas de grande porte (milhões de linhas de código) e em poucas situações sentimos efetivamente necessidade de trabalhar com diagramas. Usamos um sistema próprio automatizado de controle de versão e rastreabilidade requisito-código, além de outras ferramentas de ciclo de vida, que têm se mostrado bastante efetivas para nossas necessidades. Obs. temos 15 equipes ágeis trabalhando com uma abordagem fortemente baseada em Kanban e Scrum.”; “É um tema de grande importância pois durante o desenvolvimento a aplicação do UML é de grande ajuda. Já trabalhei em um ambiente onde a documentação é prioridade é isso faz toda diferença.”;

5. Conclusões

A pesquisa buscava entender o grau de utilização da UML no contexto dos métodos ágeis. A pesquisa apontou para 28,6% de utilização da UML no contexto dos métodos ágeis. Os diagramas mais utilizados foram: classes; caso de uso e atividade. Além desse resultado, a pesquisa produziu depoimentos que agregam valor à UML, por exemplo: “Acredito que o UML seja muito interessante em vários aspectos dependendo da sua utilização. Como exemplo posso citar um projeto em que trabalhei em que tínhamos um gerador de código que gerava toda uma aplicação partindo somente do modelo UML. Esse modelo era lido e gerávamos desde o banco de dados até os CRUDs e telas da aplicação.”. Desta forma, é possível afirmar que a UML pode ser usada em sintonia aos métodos ágeis.

Por outro lado, a pesquisa apontou que 71,4% não utilizam a UML no contexto dos métodos ágeis. O principal motivo para a não utilização foi a “A complexidade da utilização e quantidade de diagramas” e para suprir a necessidade de promover a documentação, as principais técnicas citadas foram: Relatório próprio; Histórias de usuário e BDD.

Sendo assim, é possível concluir que a UML apesar de ter aderência aos métodos ágeis e ter comprovada sua utilização, ela é majoritariamente não utilizada no contexto dos métodos ágeis.

Com esse conhecimento atual, é possível visualizar alguns fatos: i) para o mercado de trabalho: demanda de profissionais sem conhecimento aprofundado em

UML; demanda de profissionais com conhecimentos em metodologias ágeis; demanda de profissionais com conhecimentos em técnicas com História de usuário, BDD; ii) para as instituições de ensino: adaptação do currículo para suprir as demandas do mercado de trabalho, dedicando mais tempo às metodologias ágeis e suas técnicas.

A pesquisa recebeu bons *feedbacks* dos participantes, que enalteceram a iniciativa, por exemplo: “Na minha opinião é importante esse tipo de pesquisa, para que seja norteado as estruturas das disciplinas do curso com o mercado, como não trabalho diretamente com o desenvolvimento de software não tenho muito conteúdo para contribuir, porém concordo com a importância da documentação de processos, produtos, mudanças, etc. Já me deparei com diversos problemas em empresas por falta de documentação, algumas não dão o devido valor principalmente na documentação de processos, ficando a mercê de pessoas.”

Como trabalhos futuros, buscar novos meios de aumentar a base amostral, assim aumentando a participação e enriquecendo a base de dados, produzindo novos dados, atualizando a pesquisa.

6. Referências

- Booch Grady et al. UML: Guia do Usuário, tradução da segunda edição. Rio de Janeiro - Campus, 2012.
- Carvalho, D. R. e Braga, J. L. Avaliação de ferramentas de apoio a melhoria de processos de software em micro e pequenas empresas. 44º JAIIO - 16º ASSE Simposio Argentino de Ingeniería de Software. Rosário/Ar: JAIIO. 2015. p. 191-204.
- Filho, Wilson. P. P. Engenharia de Software: Fundamentos, Métodos e Padrões. Rio de Janeiro. 3ª Ed. – LTC, 2009.
- Furlan, José Davi. Modelagem de Objetos através da UML: The Unified Modeling Language. São Paulo - Makron Books, 1998.
- IBM. The IBM Rational Unified Process for System z. 2007. Disponível em: <<https://www.redbooks.ibm.com/redbooks/pdfs/sg247362.pdf>>. Acesso em: 18 Maio 2017.
- Lima, W. V.; Albuquerque, N. Scrum no Brasil. 2011. Monografia de Especialização. Universidade do Sul de Santa Catarina.
- Petre, M. UML in practice. In 35th International Conference on Software Engineering (ICSE 2013), 18-26 May 2013, San Francisco, CA, USA, pp. 722-731.
- Pressman, Roger; M, Bruce. Engenharia de Software. São Paulo. 8ª Ed. - McGraw Hill Brasil, 2016.
- Sommerville, I. Engenharia de Software. São Paulo. 9ª Ed. - Pearson, 2011.
- Vicari, Rosa Maria. Um Metamodelo UML para a Modelagem de Requisitos em Projetos de Sistemas MultiAgentes. 2012. Tese de Doutorado. Universidade Federal do Rio Grande do Sul.
- Wazlawick, Raul Sidney. Metodologia de pesquisa em Ciência da Computação. 2ª Ed. – Campus, 2014