

Fatores que influenciam a realização de análise e projeto de software: Um estudo qualitativo

Pablo José L. de Lima

Curso de Bacharelado em Sistemas de Informação – Universidade Federal da Paraíba (UFPB)
– Campus de IV
58280-990 – Rio Tinto– PB– Brasil

pablo.lima@dce.ufpb.br

Abstract. *Software has become a big need in today's market, but creating a software product is not trivial. Software engineering has several phases for software development and has emerged to streamline and ensure a quality product. One of the phases of software engineering is related to analysis and design, where models are created to facilitate the development of the final product. However, some teams do not create these models and end up jumping this phase and going straight to coding. Based on a qualitative exploratory study, this article aims to identify factors that influence the analysis and design of software based on the experience of professionals working in the area of software engineering.*

Resumo. *O software se tornou uma grande necessidade no mercado atual, mas a criação de um produto de software não é algo trivial. A engenharia de software possui várias etapas para o desenvolvimento de um software e surgiu para agilizar e garantir um produto de qualidade. Uma das etapas da engenharia de software é a de análise e projeto, onde são criados modelos que facilitam o desenvolvimento do produto final. No entanto, algumas equipes não criam esses modelos e acabam pulando essa etapa, indo direto para codificação. Este artigo tem como objetivo, a partir de um estudo exploratório qualitativo, identificar fatores que influenciam na análise e projeto de software baseado na experiência de profissionais atuantes na área de engenharia de software.¹*

1. Introdução

A prática da engenharia de software consiste em uma série de princípios, conceitos, métodos e ferramentas que devem ser considerados no planejamento e desenvolvimento de um software [Pressman e Maxim 2016]. Para entregar o software com as funções e características que satisfaçam as necessidades de todos os envolvidos, é necessária a compreensão do sistema através de diferentes perspectivas e níveis de abstração. De acordo com Appleton (1997), tal compreensão é facilitada quando o processo é baseado em padrões que ajudam a criar uma linguagem que possa ser compartilhada.

Na fase de análise e projeto de um processo de desenvolvimento de software, modelos são elaborados com o objetivo de representar as informações que o software transforma, a arquitetura e as funções que permitem a transformação, as características que os usuários desejam e o comportamento do sistema à medida que a transformação ocorra [Pressman e

¹ Trabalho de Conclusão de Curso do discente Pablo Jose Luiz de Lima, sob a orientação do docente José Adson Oliveira Guedes da Cunha, submetido ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal da Paraíba, Campus IV, como parte dos requisitos necessários para obtenção do grau de Bacharel em Sistemas de Informação.

Maxim 2016]. Basicamente, duas classes de modelos podem ser criadas: de requisitos (ou de análise), descrevendo o software no domínio da informação, funcional e comportamental, e de projeto, representando características do software que ajudam os desenvolvedores a construí-lo com eficiência: a arquitetura, a interface do usuário e os detalhes dos componentes.

Uma forma comum de modelagem de sistemas procedurais é através de fluxogramas, enquanto que a modelagem de sistemas orientados a objeto normalmente é realizada através de UML (*Unified Modeling Language*), cuja versão 2.5 [OMG 2015] apresenta 14 tipos de diagramas classificados em estruturais e comportamentais. Apesar da variedade de opções de modelagem para representar as diferentes perspectivas de um software, segundo Ambler (2002), é essencial criar modelos que sejam úteis, simples e fáceis de alterar para fins de comunicação e obtenção de feedback sobre o que está sendo construído.

Ao mesmo tempo em que modelos são importantes para facilitar a comunicação e manutenibilidade do software, tem-se que pesar a agilidade requerida para o desenvolvimento de software, influenciado por fatores técnicos, culturais e humanos [Demarco e Lister 2013]. Apesar de a análise e projeto ser uma fase intermediária entre requisitos e construção, nem sempre a mesma é planejada e executada [Forward et al., 2010], sendo a codificação realizada a partir da especificação dos requisitos. Diante disso, este trabalho teve como objetivo oferecer uma visão sobre o tema na perspectiva de profissionais da indústria de software da cidade de João Pessoa, na Paraíba, sendo elencados os fatores individuais e contextuais que influenciam a realização da análise e projeto. Para tanto, foram realizadas entrevistas semiestruturadas com profissionais de três organizações.

As demais seções deste trabalho estão estruturadas como segue: A Seção 2 apresenta os trabalhos relacionados. A Seção 3 descreve o método utilizado. Na Seção 4 são discutidos os resultados do estudo qualitativo. A Seção 5 apresenta as limitações da pesquisa e, por fim, a Seção 6 apresenta as conclusões e trabalhos futuros.

2. Fundamentação Teórica

Profissionais de vários ramos da indústria utilizam modelos gráficos para representar o que estão criando ou estudando: os químicos usam diagramas 2D e 3D para modelar moléculas e os cartógrafos usam vários tipos de mapas para representar diferentes aspectos geográficos de uma região. No contexto do desenvolvimento de software, um desenvolvedor pode usar diagramas para representar os vários aspectos de um projeto de software. Apesar de tal abstração ser necessária, nem sempre isso é feito por vários fatores.

Em um survey realizado com 113 profissionais em desenvolvimento de software, Forward et al. (2010) confirmaram a UML como a notação de modelagem dominante. Além disso, os autores concluíram que ferramentas de modelagem são usadas principalmente para criar documentação e para o projeto inicial com pouca geração de código. De acordo com os participantes da pesquisa, as abordagens centradas no modelo são mais fáceis, porém, não muito populares, já que a maioria dos participantes trabalham em ambientes centrados em código.

De acordo com um survey realizado com 3785 desenvolvedores, Gorschek et al. (2014) concluíram que o uso de modelos diminui com o aumento da experiência e aumenta com maior nível de qualificação. Além disso, os modelos são usados principalmente como um mecanismo de comunicação e colaboração em grupo, raramente são atualizados após a criação inicial e geralmente são desenhados em um quadro branco ou em papel, sem

preocupação com notação. Lange et al. (2006), em uma pesquisa com 80 arquitetos sobre o uso da linguagem UML, concluíram que a aderência ao padrão da notação é baixo.

Hutchinson et al. (2011) investigaram e documentaram um conjunto de fatores técnicos, organizacionais e sociais que influenciam no sucesso da engenharia orientada a modelos nas organizações e constataram que um dos principais fatores está relacionado a problemas no treinamento dos profissionais. Em um survey com 155 profissionais de software da indústria italiana, Torchiano et al. (2013) classificaram a modelagem como uma técnica relevante que proporciona uma melhoria na documentação, facilidade de manutenção e aumento da qualidade do produto. No entanto, exige um grande esforço para ser produzido, sendo, frequentemente, produzidos modelos não úteis devido à falta de ferramentas de suporte e falta de competência dos profissionais.

Agner et al. (2013) pesquisaram o uso de UML e abordagens orientadas a modelo para o projeto de software embarcado no Brasil. Os resultados apresentam que a maioria dos participantes são claramente conscientes do valor da modelagem, atestando a produtividade e portabilidade como as principais vantagens do seu uso, apesar de que tais modelos são gerados de forma limitada.

Apesar de se reconhecer a importância da modelagem durante a fase de análise e projeto no desenvolvimento de software, poucos modelos são produzidos ao longo do processo. Apesar de a UML ser a notação mais utilizada para modelagem, em geral, os modelos são criados sem aderência ao padrão, sendo necessário treinamento e competência técnica dos profissionais. Compreender de forma empírica o que influencia a realização de análise e projeto de software é importante não só para a indústria mas também para a academia de modo que possa se ajustar a forma como as disciplinas relacionadas à modelagem são lecionadas.

3. Método

De acordo com Dybå et al. (2005), a Engenharia de Software Baseada em Evidências deve prover a base científica para a realização de mudanças nos processos relacionados ao desenvolvimento de software. Esta pesquisa visa obter conhecimento sobre como os profissionais de desenvolvimento de software interpretam suas experiências no que diz respeito à fase de análise e projeto de software e por que certas combinações de fatores individuais e contextuais levam à elaboração de mais ou menos modelos.

Uma vez que estamos interessados em uma visão mais ampla do fenômeno, a questão de pesquisa não foi restritiva. Portanto, a principal questão da pesquisa é:

Quais fatores influenciam a análise e projeto na perspectiva de profissionais de desenvolvimento de software?

3.1 Coleta dos dados

Tendo em vista a diversidade e riqueza dos dados coletados, a amostragem teórica nesta pesquisa qualitativa básica objetivou incluir profissionais de diferentes organizações e com diferentes níveis de experiência e formação [Merriam 2009]. A unidade de análise foi o profissional em atividade durante a pesquisa. Ao todo foram realizadas 6 entrevistas. A Tabela 1 apresenta as informações dos participantes.

Tabela 1. Perfil dos participantes

ID	Formação	Anos de experiência no mercado	Função predominante	Tipo de Organização	Tamanho da Organização	Tempo da entrevista
PR01	Bacharel em Ciência da Computação	9	Arquiteto de software	Pública	Grande porte	22 minutos
PR02	Tecnólogo em Sistemas para Internet	1	Scrum Master	Privada	Pequeno porte	10 minutos
PR03	Bacharel em Sistemas para Internet	4	Gerente de Projeto	Privada	Pequeno porte	18 minutos
PR04	Bacharel em Sistemas de Informação	4	Desenvolvedor	Privada	Pequeno porte	17 minutos
PR05	Bacharel em Ciência da Computação	14	Especificador	Pública	Grande porte	21 minutos
PR06	Bacharel em Sistemas para Internet	1	Scrum Master	Privada	Pequeno porte	15 minutos

2

Os dados foram coletados através de entrevistas semiestruturadas com profissionais da indústria de software. O guia de entrevista foi composto de perguntas abertas e incluiu diferentes tipos de perguntas com o objetivo de explorar experiências e comportamentos, opiniões e valores, sentimentos, conhecimento e os antecedentes dos participantes (Apêndice A). O guia de entrevista foi refinado através de uma entrevista-piloto, sendo essa realizada com um sétimo participante, que serviu apenas para medir a duração de uma entrevista e verificar se as perguntas realmente atingem o objetivo. Tal entrevista foi descartada ao fim das medições e não foi utilizada na pesquisa.

As perguntas foram apresentadas em formato de funil, começando com questões gerais e avançando para mais específicas [Runeson e Host 2008]. As questões gerais incentivam reflexões importantes trazendo mais detalhes ao responder as questões específicas, possibilitando assim entender como experiências passadas podem influenciar preferências, sentimentos e comportamentos. Elas também servem para criar uma relação estreita e confiável entre o pesquisador e o entrevistado. Todas as questões positivas tiveram um negativo correspondente (ver P4 e P5), conforme protocolo apresentado no Apêndice A.

² <http://www.sebrae-sc.com.br/leis/default.asp?vcdtexto=4154>.

Para atender aos requisitos éticos deste tipo de pesquisa, cada participante foi informado sobre a pesquisa e seus direitos a fim de garantir a confidencialidade dos dados fornecidos, o anonimato do participante e o direito de se retirar da pesquisa a qualquer momento [Merriam 2009]. A autorização foi concedida através de uma pergunta específica sobre seu acordo para participar desta pesquisa. Algumas entrevistas foram realizadas nas próprias instalações das organizações e outras através de teleconferência ao longo do mês de outubro de 2017. Todo o áudio das sessões de entrevista foi gravado com o consentimento dos participantes para posterior transcrição.

3.2 Análise dos dados

Para a análise dos dados coletados foi empregado o método de análise de conteúdo [Bardin 1977], composto por três fases: (i) *pré-análise*; (ii) *exploração do material*; e (iii) *tratamento dos resultados, inferência e interpretação*.

A análise do conteúdo das entrevistas foi realizada por meio da Análise Categórica. Em tal abordagem, os textos são agrupados de forma análoga através de categorias, sendo uma boa alternativa quando se busca o estudo e aprofundamento de crenças, valores, opiniões e atitudes. Este trabalho foi realizado em conformidade com o método de Análise de Conteúdo Temático-Categórico elaborado por Oliveira (2008), o qual sugere a aplicação das seguintes etapas:

- **Preparação das informações:** Todas as entrevistas serão transcritas para serem posteriormente analisadas através da técnica de Análise de Conteúdo.
- **Transformação do conteúdo em unidades de registro (UR):** Nesta etapa determina-se a escolha do tipo de unidade de registro (palavra, frase ou parágrafo) que será adotada pelo pesquisador no decorrer da análise. Neste trabalho, foi definido como UR os recortes do texto avaliados como relevantes para compreensão do objeto de estudo.
- **Classificação de unidades de registro em temas:** Cada unidade de registro é associada a um ou mais temas, com o propósito de associar URs dentro de um mesmo contexto de significação. Cada tema, portanto, é composto por uma ou mais URs. Os temas emergem durante o processo de análise, ou seja, não são pré-estabelecidos.
- **Classificação de temas em categorias:** Refere-se ao agrupamento dos temas e suas respectivas unidades de registro em categorias. Cada categoria está associada a um conjunto de temas, e cada tema à apenas uma categoria, conforme ilustrado na Figura 1.
- **Descrição:** Esta etapa corresponde ao resumo numérico das etapas anteriores com o objetivo de apresentar a soma das unidades de registro encontradas no texto de cada entrevista, por tema.
- **Interpretação:** Para cada tema pertencente a uma categoria específica, o investigador deve fazer suas ponderações, a fim de elucidar a compreensão e a importância da existência do tema.

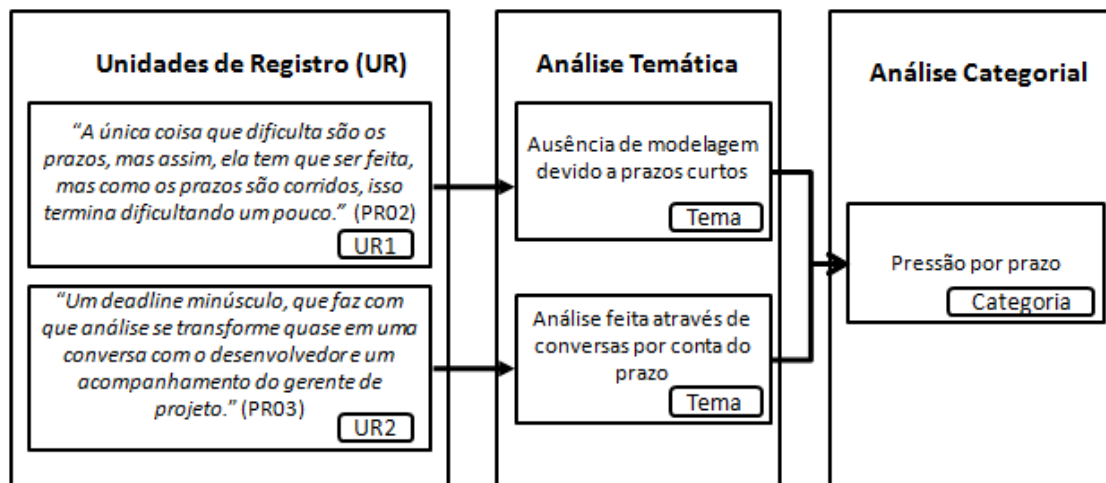


Figura 1. Exemplo de unidades de registro, temas e categoria

4. Resultados

De acordo com a análise realizada através das entrevistas com os profissionais de software, foram identificadas 3 categorias relacionadas a fatores contextuais (ambientais) e 4 categorias relacionadas a fatores individuais. O modelo gerado com as categorias está ilustrado na Figura 2.

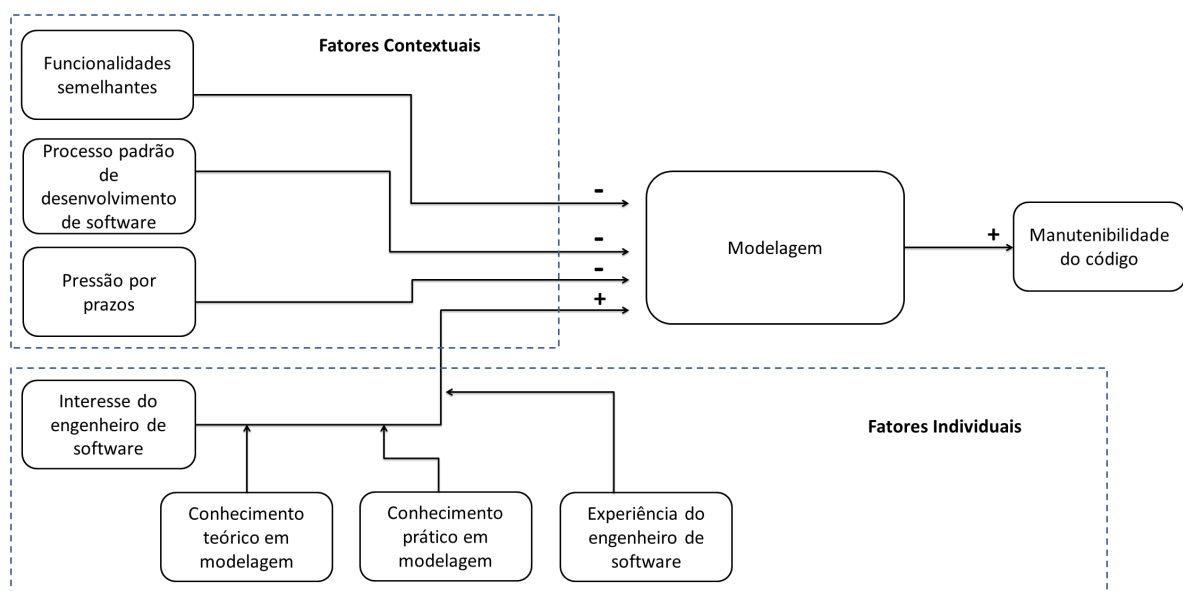


Figura 2. Modelo resultante da análise

As categorias relacionadas mais presentes nas entrevistas foram: “*Pressão por prazos*”, tendo 6 URs associadas e “*Interesse do engenheiro de software*” com 5 URs, aparecendo em um total de 15,7% das URs que foram identificadas nas entrevistas. A categoria menos presente foi “*Funcionalidades semelhantes*” tendo apenas 1 UR. A frequência com que a categoria é repetida não necessariamente deve ser levada em consideração para conclusão que uma categoria é de maior importância para a pesquisa, pois para a pesquisa qualitativa os resultados vão além de números e evidências.

As seções 4.1 e 4.2 apresentam, respectivamente, as categorias referentes aos fatores contextuais e individuais. A seção 4.3 apresenta manutenibilidade do código como consequência levantada da realização de modelagem.

4.1 Fatores contextuais

Os fatores contextuais são relacionados ao ambiente de trabalho dos profissionais entrevistados. Tais fatores afetam de forma direta a realização ou não da análise e projeto durante o processo de desenvolvimentos de software. As categorias correspondentes são: “*Processo padrão de desenvolvimento de software*”, “*Pressão por prazos*” e “*Funcionalidades semelhantes*”.

Na categoria “*Pressão por prazos*”, de acordo com os relatos dos entrevistados, percebe-se que a pressão da empresa para entrega dos projetos nos prazos estipulados e a falta de tempo implicam diretamente na não realização da análise e projeto, conforme exemplificado nas URs abaixo:

- “*É de minha responsabilidade a cobrança da modelagem, mas é de minha responsabilidade a entrega em tempo hábil. Como manter maturidade do processo de software requer tempo a modelagem também exige, colocando na balança os pacotes de entrega (dinheiro) é o “correto” a se fazer.*” (PR03)
- “*A única coisa que dificulta são os prazos, mas assim, a modelagem tem que ser feita, mas como os prazos são corridos, isso termina dificultando um pouco.*” (PR02)
- “*Prazos são apertados e sempre entra primeiro as atividades de implementações.*” (PR06)

Outro ponto que foi percebido que impacta na análise e projeto de acordo com a categoria “*Processo padrão de desenvolvimento de software*”, foi a existência de um processo padrão de desenvolvimento de softwares, sendo ele uma adaptação de algum processo de desenvolvimento que se encaixe nos hábitos da empresa.

- “*Hoje em dia praticamente não trabalhamos mais com modelos, desde que foi lançada a última versão do processo de desenvolvimento para trabalharmos com o processo ágil.*” (PR01)
- “*A gente trabalhou bastante tempo com uma espécie de RUP uma coisa até meio antiga, empresas públicas são mais difíceis de ‘acompanhar’ o mercado, pra você ver, agora depois de um ano e meio é que eles tão começando a ter uma visão mais na parte de SCRUM, e com isso não modelamos mais, agora assim, numa exigência contratual a gente utiliza o modelo antigo.*” (PR05)
- “*A empresa utiliza em sua maioria a metodologia ágil SCRUM somados a outros processos que envolvem medição, indicadores e controle de mudanças.*” (PR03)

Na categoria “*Funcionalidades semelhantes*” o entrevistado não vê a necessidade da utilização da modelagem, pois geralmente na sua empresa os projetos e as funcionalidades seguem um padrão que toda a equipe já está acostumada, sendo assim a equipe já parte diretamente para a codificação das funcionalidades que foram demandadas.

- *“Temos uma arquitetura bem distinguida, e normalmente a gente consegue encaixar um desenvolvimento dentro dessa arquitetura e sanar nossos problemas.” (PR01).*

4.2 Fatores individuais

Os fatores individuais são relacionados ao conhecimento, sentimento e experiência dos entrevistados, e como isso influencia ou não a análise e projeto. As categorias correspondentes são: *“Interesse do engenheiro de software”*, cuja relação com a realização de modelagem é moderada por *“Conhecimento prático em modelagem”*, *“Conhecimento teórico em modelagem”* e *“Experiência do engenheiro de software”*. Nesse caso, o interesse do engenheiro de software influencia a elaboração de modelos, sendo tal relação influenciada pelo conhecimento teórico e prático e pela experiência do engenheiro de software.

Através da análise das entrevistas, pudemos observar que na categoria *“Interesse do engenheiro de software”* muitos dos entrevistados, apesar de já terem tido contato com a modelagem de dados, seja de forma acadêmica ou no ambiente de trabalho, não se sentem interessados em praticá-la por vontade própria.

- *“Eu acho essa parte de modelo um pouco burocrática, na verdade o que eu gosto mesmo é de desenvolver.” (PR04)*
- *“Você tá falando com o cara que detesta modelos, eu nunca tive vontade de elaborar esses modelos.” (PR01)*
- *“Eu trabalho mais com a parte de documentação, Eu não sou assim, tão envolvido com modelo de dados.” (PR05)*

Na categoria *“Experiência do engenheiro de software”* outros fatores que influenciam na não realização da análise e projeto estão relacionados a falta de experiência dos profissionais entrevistados.

- *“Eu ainda sou inexperiente em análise e projeto, até porque depois de formado, como funcionário mesmo, vai completar um ano.” (PR02)*

Segundo os dados analisados da categoria *“Conhecimento prático em modelagem”* A falta de prática de modelagem acaba gerando uma dificuldade na de análise e projeto entre os profissionais, como foi citado, alguns tiveram um base de conhecimento em modelagem durante a formação acadêmica mas não houve muita utilização no âmbito profissional.

- *“Não é uma área na qual eu tive muita prática tirando a parte mais acadêmica” (PR06)*
- *“Eu não gosto muito de fazer diagrama de sequência, porque essa parte de fazer todas as interações, sistema, usuário, volta pro usuário, às vezes é um tipo de usuário diferente, eu acho isso muito complexo.” (PR04)*

Na categoria *“Conhecimento teórico em modelagem”* é abordado os temas que são relacionados ao conhecimento teórico em questões gerais. Nota-se que alguns entrevistados

tiveram um bom embasamento teórico durante a graduação, enquanto outros apenas tiveram uma introdução sobre as disciplinas que abordam a análise e projeto de software.

- *“Durante minha formação, devido o curso ser Tecnólogo o foco foi dado a programação as cadeiras de análise serviram apenas de apoio e “saber que existe” essa função.” (PR03)*
- *“A gente trabalhou muito essa questão de análise, de elaborar documento, elaborar modelo e propor o caso de uso, diagrama de atividade, de sequência, eu acho que assim, tudo que eu vi nessas disciplinas eu consigo usar no mercado de trabalho.” (PR04)*

4.3 Manutenibilidade do código

A modelagem tem como produto final um modelo de dados que tanto pode servir de base para a codificação, como um documento que valida as funcionalidades. O modelo gerado provém a manutenibilidade do projeto, tendo em vista que se tem um protótipo do que se foi desenvolvido tornando a rastreabilidade mais rápida, diminuindo o retrabalho, entre outros pontos.

É percebido que os entrevistados reconhecem a importância da modelagem e de um modelo que dê suporte ao desenvolvimento ou que sirva como um documento, onde se pode definir e assegurar determinadas funções.

- *“Falta de documentação formal para defender ambos os lados (desenvolvedores e analistas).” (PR03)*
- *“A análise e projeto traz benefícios para todo o projeto, a parte de codificação por exemplo, fica bem mais tranquilo de se fazer.” (PR05)*
- *“Eu acho que é uma fase do projeto de grande importância que pode gerar um grande apoio pra parte de codificação.” (PR06)*

5. Limitações da Pesquisa

Diferente do paradigma pós-positivista que busca generalizar os resultados [CRESWELL 2013], a abordagem utilizada neste trabalho, através de uma pesquisa qualitativa, tem o propósito de adquirir os conhecimentos oriundos da perspectiva dos profissionais de desenvolvimento de software entrevistados, não necessariamente com o objetivo de generalizá-los.

Uma ameaça ao estudo é o tempo relativamente curto para condução do mesmo. O estudo é complexo e demanda um espaço maior de tempo para pesquisa e análise dos dados, onde poderia-se retornar em campo e reavaliar a análise junto aos entrevistados, realizar novos questionamentos e obter novos dados.

Outro ponto é a quantidade de profissionais entrevistadas. Apesar de as informações obtidas através das entrevistas realizadas serem ricas, uma gama maior de entrevistados deixaria a pesquisa com maior embasamento de modo a atingir a saturação.

6. Conclusões e Trabalhos Futuros

Foi realizada uma pesquisa qualitativa baseada na experiência acadêmica e prática de profissionais atuantes na área de TI, que listou os pontos que influenciam positivamente ou negativamente a realização da análise e projeto de software. De acordo com a análise realizada foi criado um modelo que tem como categorias gerais os fatores contextuais, sendo eles: “*Processo padrão de desenvolvimento de software*”, “*Pressão por prazos*” e “*Funcionalidades semelhantes*”. e os fatores individuais, sendo eles: “*Interesse do engenheiro de software*”, “*Conhecimento prático em modelagem*”, “*Conhecimento teórico em modelagem*” e “*Experiência do engenheiro de software*” e também a categoria “*Manutenibilidade do código*” como consequência da modelagem.

Foi possível identificar vários pontos de pesquisas durante a análise das entrevistas, como a falta de interesse em muito dos profissionais em realizar modelagem e a pressão sofrida pelos prazos que são provenientes de fatores internos ou externos.

Como trabalho futuro, sugere-se voltar a campo e validar a análise, colher mais informações a partir das análises feitas, reaplicar o método de entrevista com outros profissionais de TI e abranger mais a pesquisa com outras empresas, tendo em vista uma diversificação maior de dados e fatores que corroborem para a realização ou não da análise e projeto.

Referências

- Agner, L. T. W. et al. A Brazilian survey on UML and model-driven practices for embedded software development. *Journal of Systems and Software*, v. 86, n. 4, p. 997-1005, 2013.
- Ambler, S. Agile modeling: effective practices for extreme programming and the unified process. John Wiley & Sons, 2002.
- Appleton, B. Patterns and software: Essential concepts and terminology. 1997.
- Bardin, L.. Content analysis. Editions Lisbon, 1977.
- Creswell, J. W. Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications, 2013.
- Demarco, T., e Lister, T. Peopleware: productive projects and teams. Addison-Wesley, 2013.
- Dybå T., Kitchenham, B. A., e Jørgensen, M. Evidence-Based Software Engineering for Practitioners, *IEEE SOFTWARE*, Vol. 22, No. 1, pp 58-65, 2005.
- Forward, A., e Badreddin, O., e Lethbridge, T. C. Perceptions of software modeling: a survey of software practitioners. In: 5th workshop from code centric to model centric: evaluating the effectiveness of MDD (C2M: EEMDD). 2010.
- Gorschek, T., e Tempero, E., e Angelis, L. On the use of software design models in software development practice: An empirical investigation. *Journal of Systems and Software*, v. 95, p. 176-193, 2014.
- Hutchinson, J. et al. Empirical assessment of MDE in industry. In: Software Engineering (ICSE), 2011 33rd International Conference on. IEEE, 2011. p. 471-480.
- Lange, C. F. J., Chaudron, M. R. V., e Muskens, J. In practice: UML software architecture and design description. *IEEE software*, v. 23, n. 2, p. 40-46, 2006.
- Merriam, S. B. Qualitative Research: a Guide to Design and Implementation, San Francisco, CA: Jossey-Bass, 2009.

Oliveira, D. C. Análise de conteúdo temático-categorial: uma proposta de sistematização. Rev. enferm. UERJ, 16(4):569–576, 2008.

OMG. Unified Modeling Language version 2.5 , 2015.

Pressman, R, e Maxim, B. Engenharia de Software-8ª Edição. McGraw Hill Brasil, 2016.

Runeson, P. e Host, M. Guidelines for conducting and reporting case study research in software engineering. Empirical Software Engineering, Vol. 14, No. 2, p. 131-164, 2008.

Torchiano, M. et al. Relevance, benefits, and problems of software modelling and model driven techniques—A survey in the Italian industry. Journal of Systems and Software, v. 86, n. 8, p. 2110-2126, 2013.

Apêndice A - Protocolo da Entrevista com Profissionais de Desenvolvimento de Software

Apresentação • Auto-apresentação • Agradecimento ao participante • Permissão para gravar o áudio da entrevista Objetivo da Entrevista O objetivo desta pesquisa é entender os aspectos relacionados à análise e projeto em projetos de software. Introdução Todas as informações fornecidas nesta entrevista serão tratadas como confidenciais. Apenas a equipe de pesquisa terá acesso às informações fornecidas. Em particular, nenhuma pessoa direta ou indiretamente ligada a sua empresa terá acesso às informações fornecidas nesta entrevista e em nenhuma outra fase da pesquisa. O conteúdo das entrevistas será utilizado apenas para fins da pesquisa acadêmica, não tendo assim nenhuma influência na avaliação do funcionário no desempenho das suas atividades na organização. A entrevista será gravada para posterior documentação. Ao final da pesquisa, os dados serão publicados em eventos de natureza acadêmica, mas os nomes das pessoas envolvidas serão omitidos nas respectivas publicações. Sua participação nesta pesquisa é voluntária e você pode decidir não participar ou se retirar da pesquisa a qualquer momento. Caso você decida não participar, não receberá nenhuma sanção ou penalidade. Você concorda em participar desta pesquisa? Identificação do entrevistado As informações a seguir serão utilizadas caso a equipe de pesquisa precise entrar em contato com você no futuro para esclarecimentos sobre a entrevista. Por favor, diga seu nome e e-mail.	
P1. [Antecedente] Fale um pouco de você: sua formação, idade, trajetória profissional. <i>Sondagem:</i> Quais funções você exerceu em projetos de software? <i>Sondagem:</i> Como foi sua experiência com análise e projeto durante sua formação? P2. [Antecedente] Há quanto tempo você trabalha nesta empresa? P3. [Experiência] A organização utiliza algum processo de desenvolvimento de software? <i>Sondagem:</i> Descreva-o em linhas gerais. P4. [Opinião] O que a sua organização oferece ou faz para estimular a análise e projeto de software? P5. [Opinião] O que a sua organização faz (e/ou que não deveria fazer) que dificulta a análise e projeto de software? P6. [Experiência] Fale sobre o projeto em que trabalha atualmente. <i>Sondagem:</i> Quais são as principais dificuldades que vocês enfrentam no projeto em que trabalha? P7. [Opinião] Quais modelos são mais elaborados em seu projeto? <i>Sondagem:</i> Quais são os métodos e ferramentas utilizados? P8. [Experiência] Qual influência você exerce na modelagem em seu projeto? P9. [Experiência] Que problemas você vivenciou pela falta de modelos que apoiem a codificação?	P10. [Experiência] Em que situações a equipe deixa de elaborar algum modelo? P11. [Sentimento] Quais tipos de modelo você se sente mais à vontade para elaborar? Por que? P12. [Sentimento] Quais tipos de modelo você se sente menos à vontade para elaborar? Por que? P13. [Opinião] Como você descreveria um colega que desempenha bem as atividades relacionadas à análise e projeto de software? P14. [Opinião] Como você descreveria um colega que desempenha mau as atividades relacionadas à análise e projeto de software? P15. [Sentimento] Como você avalia o seu desempenho na análise e projeto de software? P16. [Opinião] Qual a importância da análise e projeto? P17. Você gostaria de adicionar alguma informação ou observação que não foi perguntada, mas que você considera importante para a análise e projeto de software?