

Perceptions of 395 Developers on Software Architecture's Documentation and Conformance

Izabela Melo*, Gustavo Santos†, Dalton Dario Serey* and Marco Tulio Valente‡

* Universidade Federal de Campina Grande, Campina Grande, Paraíba, Brasil

Emails: izabela.vmelo@gmail.com, dalton@computacao.ufcg.edu.br

† Université de Lille, CNRS, Centrale Lille, INRIA, UMR 9189, CRIStAL, Lille, France

Email: gugajansen@gmail.com

‡ Universidade Federal de Minas Gerais, Belo Horizonte, Minas Gerais, Brasil

Email: mtov@dcc.ufmg.br

Abstract—Since the 90's, a major effort has been applied in the academy to study and evolve aspects related to Software Architecture. The significant rise in the number of publications in the last few years shows how much importance and concern academia gives to the subject. However, previous studies have pointed out that when it comes to software architecture concepts and practices, there is a gap between industry and academy. Industry professionals do not know/use what is proposed by academic researchers. From the academic perspective, the theoretical foundations about software architecture, architectural documentation and architectural conformance checking are well developed. Then why industry seems not to follow them? What aspects does the academy miss about the problems in practice? What is the practical/industrial context about software architecture's definition, documentation and verification? Aiming to answer these questions, we performed a 3-step qualitative study. Firstly, we applied an exploratory survey to 58 professionals to better understand the state of the practice on software development and architecture and to give us an initial comprehension of software architecture concepts, practices and problems from the point of view of the industry. Secondly, we interviewed 14 top-professionals chosen among those who participated in the exploratory survey. In this phase we intended to deeply understand the professionals' perceptions on the subject. We used thematic analysis to assess the narratives, to form a base for our understanding and to formulate our comprehension as a series of statements. Finally, in a third phase, we performed a confirmatory survey by means of which we were able to confirm the findings. We sent the survey to GitHub users with public e-mails and more than 100 followers, getting 7.74% of response rate (337 respondents). From our study, we confirmed that there is not a single and unified definition and understanding of the term "software architecture" among software developers. In particular, we observed that the definition depends on several factors, such as the professional's experience and the company in which he/she is currently working. The architecture is sometimes documented but, most of the time, it is incomplete, informal and outdated. Finally, we observed that while there is a general perception of the relevance of architectural conformity and compliance of software, activities to check architectural conformance and compliance are seldom adopted — not to mention the use of automated tools and procedures.

I. INTRODUÇÃO

A arquitetura de software é um dos artefatos mais importantes no ciclo de vida de um sistema [6][7]. Ela interfere nos objetivos de negócios, objetivos funcionais e na qualidade do sistema. Atualmente, existem diversas definições na literatura para o termo "Arquitetura de Software" [8]. Em 2002, havia, no site do SEI (Software Engineering Institute), mais de 90 (noventa) definições coletadas para o termo em questão [9].

O aumento significativo das publicações nas últimas décadas [10] demonstra a importância e a preocupação que a academia tem em relação ao tema, incluindo estudos sobre linguagens para documentação arquitetural [12][13][14][15], técnicas para visualização de arquitetura de software [1][16], abordagens para reconstrução de arquitetura de software [17] e até sobre o ensino da área nos cursos de graduação [18]. Os diversos estudos apontam que é preciso verificar a arquitetura de software para evitar a Erosão Arquitetural [11], ou seja, a perda da estrutura do sistema. Com isso, a academia propôs muitas ferramentas com o objetivo de auxiliar a verificação de conformidade arquitetural.

Na literatura, existem indícios que linguagens como UML, que são bastante difundidas no meio acadêmico, são pouco utilizadas no meio industrial [19]. Estudos mostram também que é necessário entender melhor o que os arquitetos de software, na prática, necessitam [14][20]. Com isso, surgem questionamentos sobre o real uso que ambientes industriais de desenvolvimento de software fazem de aspectos relacionados à Arquitetura de Software.

Com o objetivo de revelar e melhor entender esses questionamentos, realizamos um estudo quali-quantitativo sobre a percepção dos conceitos de arquitetura de software por profissionais da indústria. Entender como os profissionais da área tratam este assunto contribui com o fornecimento de diretrizes para as pesquisas que estão sendo desenvolvidas pelos pesquisadores. Formalizamos e organizamos este estudo para responder quatro questões de pesquisa relacionadas à: (i) definição de arquitetura de software, (ii) documentação

arquitetural, (iii) verificação de conformidade arquitetural e (iv) às ferramentas de auxílio a esta atividade de verificação. O estudo foi realizado com 395 profissionais da indústria, onde 58 participaram da primeira e segunda etapas e 337 participaram da última etapa.

Constatamos que, assim como na academia existem diversas definições para o termo “Arquitetura de Software”, na indústria também não há consenso. A definição na prática depende de diversos fatores como o conhecimento e experiência de cada profissional, de sua origem acadêmica e como aprenderam a teoria.

Também pudemos notar que a forma de documentação arquitetural varia muito entre empresas e profissionais. Na maioria dos projetos, a documentação arquitetural é informal, incompleta e/ou desatualizada. Normalmente, documentos escritos em linguagem natural são utilizados como forma de documentar a arquitetura. Sabendo que, na academia, frequentemente são pesquisadas e propostas novas linguagens formais e maneiras de organizar uma documentação arquitetural, nos questionamos o motivo pelo qual a indústria prefere documentos em linguagem natural ao invés das propostas acadêmicas.

Os dados também nos permitem afirmar que a atividade de verificação de conformidade arquitetural, quando realizada, é conduzida, na maioria das vezes, de forma manual. As ferramentas de apoio à verificação de conformidade arquitetural são pouco conhecidas e pouco utilizadas na indústria.

Por fim, nossos dados nos permitem constatar que a falta de um profissional responsável pela arquitetura de software, sua documentação e verificação é um dos fatores da falta de valorização dessas atividades.

O restante deste artigo está organizado como descrito a seguir. A Seção II apresenta a descrição do estudo, a Seção III apresenta os Resultados e Discussões, a Seção IV apresenta os Trabalhos Relacionados, a Seção V apresenta as Ameaças à Validade e, por fim, a Seção VI apresenta as Conclusões do estudo.

II. DESCRIÇÃO DO ESTUDO

Realizamos um estudo inspirado em uma metodologia proposta por Singer et al. [5]. O estudo foi dividido em três etapas: survey exploratório, entrevistas e survey confirmatório. Tivemos como principal objetivo responder às seguintes questões de pesquisa:

- RQ1: Qual é a visão da indústria sobre a definição de “arquitetura de software”?
- RQ2: Qual a visão da indústria com relação a documentação arquitetural?
- RQ3: Qual a visão da indústria com relação a verificação de conformidade arquitetural?
- RQ4: Como é a relação da indústria com as ferramentas de apoio a verificação de conformidade arquitetural existentes?

Com as três etapas propostas temos um estudo qualitativo (ou misto), ou seja, uma combinação de métodos qualitativos e quantitativos. Dessa forma, realizamos a Triangulação Metodológica dos Dados [4], ou seja, capturamos

os mesmos dados de diferentes formas para que as análises fossem cruzadas e validadas. Este método misto aumenta a confiabilidade nos resultados obtidos e a validade do estudo, uma vez que capturamos as informações de formas diferentes.

A. Survey Exploratório

A primeira etapa foi um survey exploratório cujo objetivo principal foi ter uma noção, em alto nível, de como o contexto de arquitetura de software está inserido, particularmente, no desenvolvimento de sistemas reais. Questões envolvendo o conceito de arquitetura na prática dos profissionais, frequência e forma de documentação e verificação de conformidade arquitetural, bem como os motivos para a não realização dessas atividades foram incluídas neste survey. As perguntas deste survey podem ser consultadas no nosso material online.¹

A amostra de candidatos a respondentes do survey foi construída a partir de uma lista de arquitetos (quando existia este papel) e *core developers* de empresas e projetos presentes em nossa rede de contatos e das companhias Fraunhofer, Oracle, Arch-Linux e Linux Kernel. Além disso, enviamos o survey para duas listas no LinkedIn voltadas para arquitetos de software e para as listas de desenvolvimento do OpenStack e do Eclipse. Para complementar a lista de candidatos, utilizamos o método *snowball*, solicitando sugestões, dos próprios respondentes, de profissionais que pudessem responder nossa pesquisa. No total, enviamos o questionário exploratório para 149 profissionais, dos quais 47 foram indicados pelos próprios respondentes do survey.

As respostas quantitativas foram sumarizadas e as respostas qualitativas foram separadas em grupos. Como exemplo de grupo podemos citar: “profissionais que realizam verificação de conformidade arquitetural sem suporte de ferramentas”. Após essa organização, as respostas foram codificadas com o objetivo de facilitar nossa análise e dar uma melhor visão do contexto de nossos respondentes com relação a nossa pesquisa.

Obtivemos uma taxa de resposta de 24,1%, o que corresponde a 36 profissionais. Além disso, obtivemos 22 respostas das 4 listas de discussões para as quais enviamos o survey. Portanto, no total, tivemos 58 respondentes. Estes, têm, em média, 12,34 anos de experiência em Tecnologia da Informação, onde o mais experiente tem 30 anos e o menos experiente tem 7 meses. A Figura 1 apresenta a distribuição do tempo de experiência dos 58 respondentes do survey exploratório. Estes respondentes também possuem experiência, principalmente, em Java (54 desenvolvedores), JavaScript (38), C++ (32), C (26), C# (22) e Python (22).

Com esta etapa, pudemos ter uma primeira percepção com relação à adoção de conceitos relacionados a arquitetura de software, nos dando apoio para planejar e executar a etapa de entrevistas.

B. Entrevistas

A segunda etapa consistiu de entrevistas e teve o objetivo de aprofundar o conhecimento e entendimento sobre as principais

¹<https://github.com/izabelavmelo/sbcars2016>

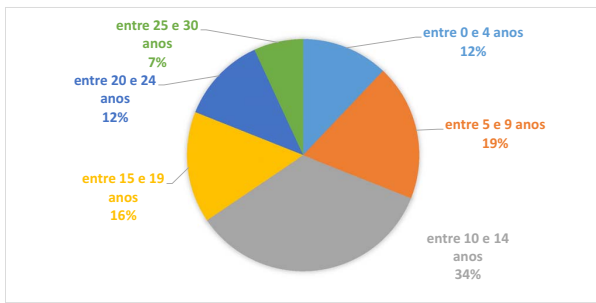


Figura 1: Tempo de experiência dos 58 respondentes do survey exploratório.

questões de pesquisa, colhendo detalhes e aspectos que não foram capturados no survey exploratório. Foram feitas, em média, 13 perguntas, envolvendo questões de definição, documentação, verificação, frequência e forma do desenvolvimento dessas atividades, assim como também a importância dada a elas. As perguntas foram formuladas a partir dos resultados da etapa anterior.

Para selecionar os participantes desta etapa, separamos os respondentes do survey exploratório que afirmaram ter desempenhado a função de arquiteto ou designer de software (49 respondentes). Destes, eliminamos o primeiro quartil (25%) menos experiente, resultando em 35 profissionais, dos quais 14 se voluntariaram e participaram desta segunda etapa. Portanto, obtivemos uma taxa de resposta de, aproximadamente, 40%.

Todas as entrevistas foram gravadas e transcritas com a autorização dos entrevistados. Elas tiveram duração média de 26 minutos, onde a entrevista mais duradoura foi de 46 minutos e a menos duradoura de 13 minutos. Os entrevistados tinham experiência média de 7 anos como arquitetos de software e trabalhavam em empresas de pequeno a grande porte no Brasil, Argentina, Uruguai, Estados Unidos da América, Alemanha, Áustria, Hungria e Reino Unido.

A análise qualitativa das entrevistas foi realizada de acordo com a Análise Temática proposta por Braun & Clarke [2]. A partir da Análise Temática, é possível identificar padrões de respostas e organizá-las de acordo com temas em comum. Utilizamos a ferramenta NVivo [23] para suporte da análise, e encontramos cinco grandes temas: Arquitetura de Software no geral, Documentação arquitetural, Verificação de conformidade arquitetural, Violações arquiteturais e Ferramentas de apoio à verificação de conformidade arquitetural. Por restrições de espaço, incluímos nesse artigo as discussões dos temas. O roteiro de entrevista e o modelo hierárquico de temas encontrado nesse trabalho está disponível no nosso material online.

C. Survey Confirmatório

Com o survey confirmatório, última etapa deste trabalho, tivemos como objetivo confirmar nossas principais conclusões acerca do estudo que realizamos. Com esta etapa, pudemos quantificar a taxa de concordância dos profissionais sobre as

principais conclusões das etapas anteriores, e assim servindo de validação das respostas obtidas anteriormente.

Utilizamos a escala Likert na maior parte do questionário, onde as perguntas são respondidas escolhendo-se um valor entre 1 e 5, de acordo com o nível de concordância do respondente com relação à nossa afirmação. Em alguns casos, foi necessário modificar a escala de resposta para termos uma noção de frequência, por exemplo, de “sempre” até “nunca”.

Enviamos nosso questionário para 4352 profissionais do GitHub que possuíam mais de 100 seguidores e e-mails disponíveis. Obtivemos 337 respostas, o que nos dá uma taxa de resposta de 7,74%. Os respondentes eram de 47 países diferentes e afirmaram trabalhar ou já terem trabalhado com 71 linguagens diferentes. A Figura 2 mostra as principais linguagens citadas pelos respondentes desta etapa. A Figura 3 mostra a distribuição do país de residência dos 337 respondentes do survey confirmatório. Os principais países dos respondentes são: EUA (38%), Brasil (8%), China (7%), Alemanha (6%), Reino Unido (4%), Canadá (4%) e França (3%). Consideramos que nossa amostra é bastante diversificada, tanto com relação à localização, quanto a experiência em linguagens de programação.

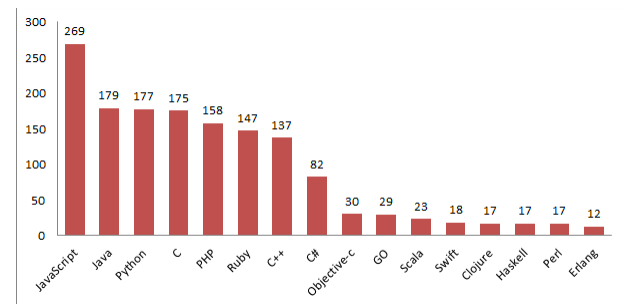


Figura 2: Principais linguagens citadas pelos 337 profissionais que responderam ao nosso survey confirmatório.

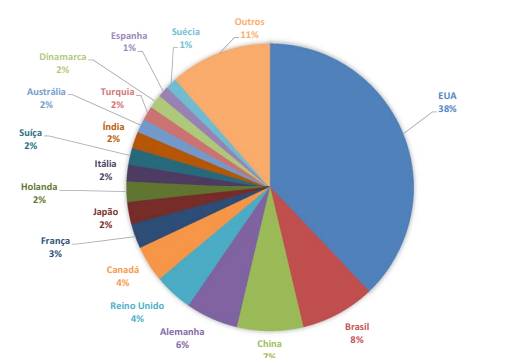


Figura 3: 337 profissionais de 47 países diferentes responderam ao nosso survey confirmatório.

III. RESULTADOS E DISCUSSÕES

Após a análise das duas primeiras etapas (survey exploratório e entrevistas), as principais conclusões foram submetidas à avaliação dos nossos respondentes do survey confirmatório, que validaram a maioria das afirmações que propusemos. Com isso, temos uma maior confiabilidade em realizar algumas considerações e revelar alguns indícios a partir de todas as respostas. Além da confirmação com a última etapa, a triangulação de dados que utilizamos em nossa pesquisa também nos dá uma maior confiança nos resultados obtidos.

No entanto, também temos consciência de que nosso estudo não pode ser generalizado e não contempla uma percepção completa do ambiente estudado. Porém, nossas conclusões foram quantificadas pela última etapa, a partir da taxa de concordância dos profissionais com relação às nossas principais conclusões, e podem ser visualizadas nas Figuras 5 e 7.

Nas próximas subseções abordamos cada questão de pesquisa proposta nesse trabalho, apresentando e analisando os resultados encontrados.

A. RQ1: Qual é a visão da indústria sobre a definição de “arquitetura de software”?

A visão da indústria, segundo nossos respondentes e entrevistados, é de que não existe uma única definição para o termo “arquitetura de software”. A Figura 5a (terceira pergunta) mostra que cerca de 80% dos respondentes do survey confirmatório concordam com esta afirmação. Nossos dados nos permitem concluir que a forma como este termo é utilizado depende de vários fatores, dentre eles a cultura da empresa, o profissional e o projeto. Inclusive, foi citado por alguns respondentes que os termos que envolvem esta área podem variar de lugar para lugar, possibilitando interpretações diferentes para o mesmo termo.

Além disso, como é possível ver na Figura 4, diferentes aspectos podem ser considerados como parte da arquitetura, tais como, aspectos estruturais, dinâmicos, de infra-estrutura, do domínio do problema, dentre outros. Os aspectos mencionados como parte da arquitetura dependem de alguns fatores como experiência e visão do profissional responsável, tamanho do projeto e objetivo com a arquitetura. Consideramos naturais esses achados e já os esperávamos, uma vez que também não existe consenso sobre essa definição no meio acadêmico.

Apesar da falta de consenso na definição de arquitetura de software tanto na academia como na indústria, identificamos que, independente dos aspectos que são considerados como parte da arquitetura, uma definição que pode ser tomada como universal é que se trata do conjunto central de decisões sobre o sistema em desenvolvimento. Esta afirmação parece ser a mais aceita tanto na academia quanto na indústria.

B. RQ2: Qual a visão da indústria com relação a documentação arquitetural?

Os profissionais consultados, de uma forma geral, consideram que documentar a arquitetura é uma atividade importante. Na Figura 5a (primeira pergunta) podemos observar que mais de 85% dos respondentes concordam com esta afirmação.

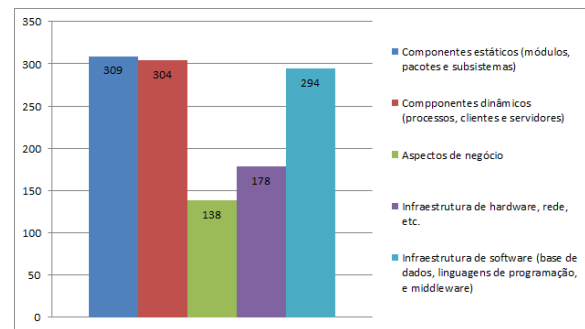


Figura 4: Aspectos que fazem parte de uma arquitetura de software.

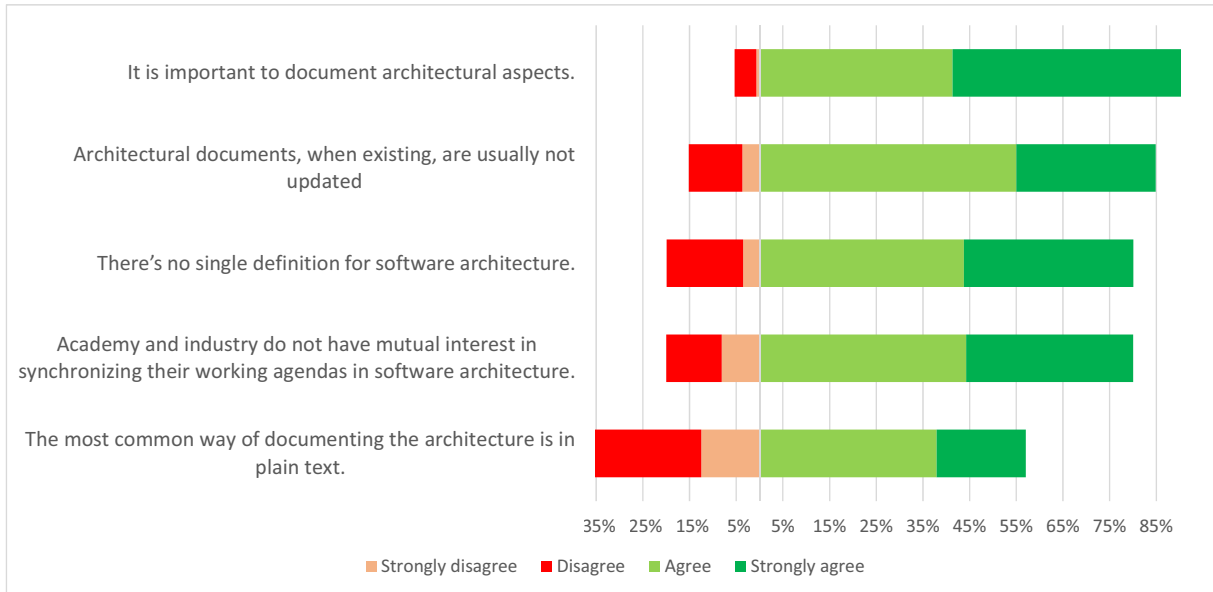
Essa atividade, segundo os profissionais participantes de nosso estudo, auxilia a manutenção e o entendimento do sistema, evita a perda de conhecimento sobre o sistema e organiza e armazena as decisões e suas justificativas.

Um aspecto que merece menção é que, pelas ponderações dos respondentes, é comum considerarem que nem todos os projetos requerem uma arquitetura precisamente definida e bem documentada. A Figura 5b (primeira pergunta) mostra que quase metade dos respondentes do survey confirmatório consideram que a documentação arquitetural não precisa ser feita de forma detalhada nunca ou apenas em alguns projetos. Para a maioria dos respondentes, dependendo do projeto, é bastante aceitável que existam documentações informais, incompletas ou desatualizadas (segunda pergunta da Figura 5a). Os sistemas críticos, por outro lado, exigem documentações rigorosas, uma vez que sua falha pode causar grandes problemas.

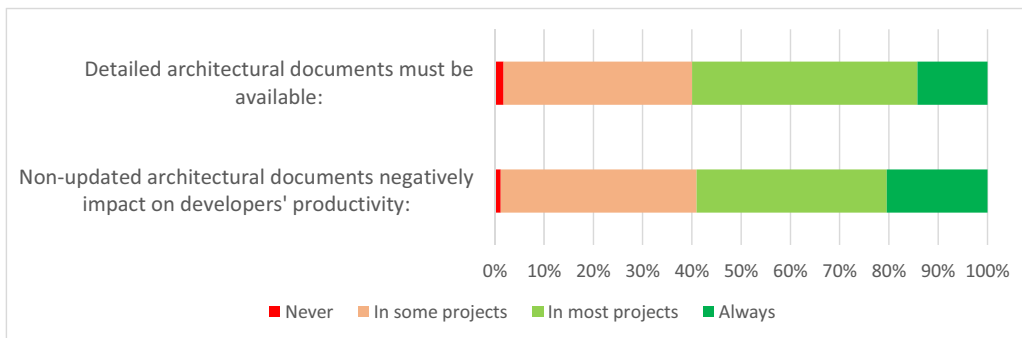
Além disso, os principais motivos para produzir documentações incompletas, informais e/ou desatualizadas são: (i) a baixa prioridade atribuída à atividade; (ii) ser considerada uma atividade de pouco valor e pouco atrativa por parte dos desenvolvedores; (iii) a descrença de que a documentação pode trazer benefícios; (iv) por considerar desnecessário em aplicações triviais, com metodologias ágeis, com uma equipe pequena, ou em sistemas de código aberto; (v) e, principalmente, o pouco tempo alocado para atividades de arquitetura no processo de desenvolvimento. É possível visualizar alguns desses motivos na resposta do entrevistado E14:

“Primeiramente, sempre se tem mais trabalho a se fazer do que tempo, tempo é um recurso bem caro [...] documentar é muito chato e as pessoas não gostam disso, elas não vêem isso como algo produtivo ou algo de valor, que nem codificar, então isso sempre acaba sendo a última prioridade e competindo com aquelas coisas que precisam de mais tempo.” (E14)

A partir dos dados podemos concluir que cada empresa tem sua forma de documentar, tanto em formatos de documentos utilizados quanto na notação utilizada. Consideramos esse aspecto natural, uma vez que existem diversas opções,



a: Escala Likert.



b: Escala de frequência.

Figura 5: Afirmações sobre definição e documentação arquitetural (RQ1 e RQ2).

linguagens e ferramentas propostas pela academia. Percebemos, porém, que a maioria dos respondentes utiliza texto em linguagem natural como principal forma de documentação. A Figura 5a (última pergunta) mostra que mais de 55% dos respondentes concordam com esta afirmação.

Concluimos, assim, que a indústria vê a documentação como uma atividade importante para alguns tipos de sistemas e.g., sistemas críticos; porém, por diversos fatores, a maioria não possui uma documentação de seus sistemas. Além disso, quando existe documentação, a maioria é escrita em linguagem natural.

C. RQ3: Qual a visão da indústria com relação a verificação de conformidade arquitetural?

De acordo com nosso estudo, a visão da indústria, de uma forma geral, é que a verificação de conformidade arquitetural é importante para a qualidade do código, porém, não é uma

atividade suficiente. A Figura 7b (primeira pergunta) mostra que quase 50% considera importante sempre ou quase sempre verificar a arquitetura e a Figura 7a (primeira pergunta) mostra que mais de 85% dos respondentes acreditam que esta atividade não é suficiente para garantir a qualidade do código.

Além disso, concluímos que nem sempre são realizadas atividades com esse propósito por diversos motivos. Dentre eles, a falta de crença de que é possível calcular ou derivar a arquitetura a partir do código fonte de um sistema, para aplicar algoritmos que façam a verificação completa da conformidade. Podemos observar isso nessa resposta do entrevistado E4:

“Anyway, I believe that architecture cannot be ‘calculated’, or best said, I think that it is not factible to apply any algorithm or suite of algorithms and techniques to verify architecture compliance” (E4)

O entrevistado E1 apontou como um dos motivos para a não realização da verificação de conformidade arquitetural a

falta de necessidade desta atividade no projeto:

“It’s depending of the needs. Sometimes I don’t need to do that” (E1).

No survey exploratório, dos 58 respondentes, 31 afirmaram não realizar verificação de conformidade arquitetural. Os motivos citados como justificativa da não realização desta atividade podem ser vistos na Figura 6. Na resposta do entrevistado E7 é possível perceber que o tamanho da empresa e seus sistemas desenvolvidos também interferem em realizar ou não esta atividade:

“Or in smaller companies it can be skill. In smaller companies, architecture conformance checking is skill of the software architect. The software architect has an intention, he has a goal he wants to reach, and if he is clever, he is not going to spread out all of the goal and all of the intention at once. He keeps it in his head, and he constantly checks the work done by people who implement the architecture against his goal. Against his intention [...] In large companies you have a process for that. Large companies, specially in Aerospace you have standards, specification documents that proscribe, that impose certain architectural style. And you also have a process to make sure that your still in conformance with standard specifications. So architectural conformance checking in a large company very often is checking the respect of standards.” (E7)

Em sua maioria, os motivos pelos quais os profissionais não realizam verificação de conformidade arquitetural são os mesmos relatados para justificar uma documentação arquitetural ruim. Consideramos esse aspecto como algo natural, uma vez que observamos as opiniões dos mesmos profissionais sobre duas atividades que impactam diretamente a arquitetura: documentação e verificação de conformidade.

Além disso, os respondentes concordam que violações arquiteturais impactam negativamente na produtividade do desenvolvimento do sistema, o que já foi confirmado na academia [12][21][22]. Na Figura 7a (sexta pergunta) é possível observar que mais de 75% dos respondentes consideram essa afirmação verdadeira.

Porém, mesmo a maioria dos profissionais envolvidos neste estudo considerando importante documentar a arquitetura e verificá-la, os profissionais não demandam muito esforço e tempo para a realização precisa dessas atividades. Essa constatação, unido a algumas respostas, como por exemplo dos Respondentes 51 e 58 (survey exploratório), nos dá a impressão de que documentar e verificar a arquitetura de software só serão atividades com alta prioridade para todos os profissionais se os problemas causados pelas violações arquiteturais forem muito graves.

“A menos que um problema surja por causa de violações arquiteturais, todos assumem que a arquitetura está sendo seguida” (Respondente 51).

“Uma das causas é a falta de ferramentas automáticas que façam essas verificações e se integrem

bem com o arcabouço de desenvolvimento que já utilizamos. Isso reduziria o custo de se fazer essa verificação, que é o segundo problema. A terceira causa é que quebras de regras arquiteturais não são (ainda) um problema regular e/ou danoso o suficiente para demandar uma verificação rotineira” (Respondente 58).

Por fim, para os que realizam a atividade de verificação, concluímos que não é comum utilizar ferramentas de apoio a essa atividade. Quando são utilizadas, trata-se de ferramentas construídas pela própria indústria. Quando é realizada, há uma preferência por realizar a atividade manualmente e os profissionais possuem confiança em seus métodos de checagem manual. A Figura 7a (último item) mostra que mais de 70% afirmam que quando a verificação é aplicada, é feita sem o suporte de ferramentas. Na Figura 7a (terceira pergunta) é possível observar que mais de 85% confirmam que as ferramentas de apoio a esta atividade raramente são utilizadas e, na Figura 7b (segunda pergunta), mais de 70% afirmaram que as ferramentas quase nunca devem ser utilizadas.

Concluímos que a verificação de conformidade arquitetural é considerada uma atividade importante para a qualidade do software, porém não é realizada com frequência. Quando realizada, os profissionais preferem utilizar métodos manuais, uma vez que as ferramentas não possuem todos os requisitos que cada um considera importante.

D. RQ4: Como é a relação da indústria com as ferramentas de apoio a verificação de conformidade arquitetural existentes?

A indústria, de acordo com nossos estudos, conhece e utiliza pouco ferramentas de verificação de conformidade arquitetural. A Figura 7a (terceira pergunta) mostra que cerca de 85% dos respondentes concordam com esta afirmação. Nossos dados nos permitem concluir que existe uma falta de interesse ou necessidade por parte de muitos profissionais em encontrar e utilizar essas ferramentas, como podemos ver na resposta do entrevistado E4:

“Ok, I don’t know any tool because I’ve never tried to find it.” (E4)

Existe também uma resistência em utilizar as ferramentas. Como é possível ver na Figura 7a (quinta pergunta), mais de 80% dos respondentes concordam com essa afirmação. Os profissionais que conheciam alguma ferramenta do gênero, em sua maioria, afirmaram que as ferramentas existentes não são fáceis de utilizar e não conseguem capturar todos os aspectos que consideram importantes da arquitetura de um sistema. É possível observar isso na resposta do entrevistado E7:

“I think that the eye of a very experienced software architect, most of the time, can and will catch most of the violations. And that trusting 100% in a tool to catch all the violations, first is going to be very costly, is going to cost you a huge amount of resources and money, and second, I’m not sure if it can be done, if a tool can really catch all the

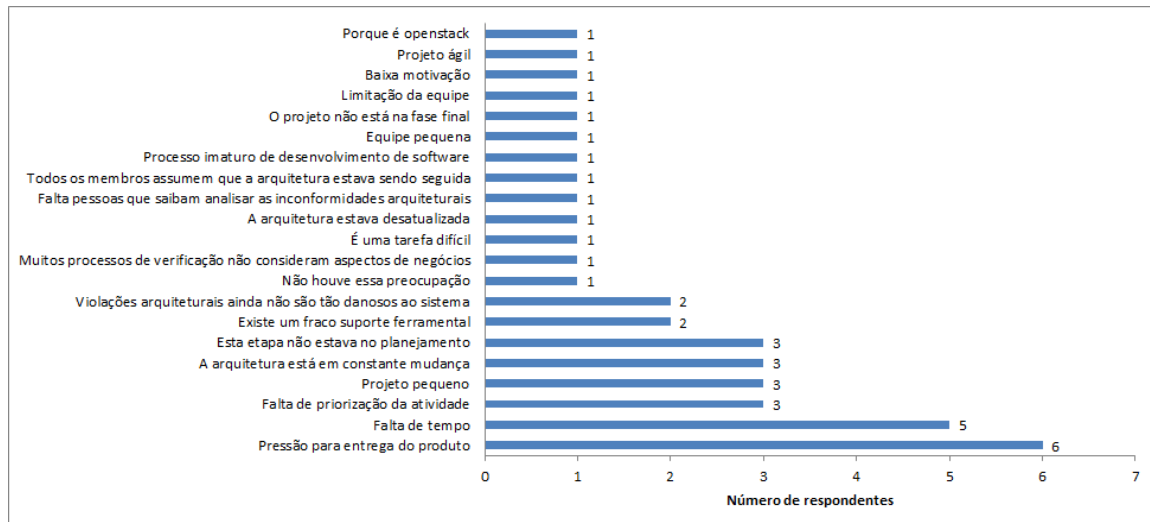


Figura 6: Justificativas dadas por 31 respondentes do survey exploratório para a não realização de atividades de verificação de conformidade arquitetural.

violations. So I would rather trust the human eye of a very experienced software architect and a tool than only a tool” (E7)

Na Figura 7a (quarta pergunta) observamos que mais de 85% dos respondentes afirmam que as ferramentas existentes são complexas e requerem documentações detalhadas. Além disso, alguns profissionais reclamaram da falta de flexibilidade das ferramentas, a exemplo do entrevistado E4:

“My software I said, I already told you, that this checking is kind of crafty. I mean, it’s made by hand. I don’t use any tool, because tools are not flexible” (E4)

Por fim, notamos que alguns respondentes afirmaram perder muito tempo na configuração das ferramentas. Assim, refletimos que a união dessas justificativas sejam os motivos para que exista uma certa resistência, por parte de alguns profissionais, em utilizar esses tipos de ferramentas.

E. Outras discussões

Nosso estudo nos trouxe conclusões que vão além dos limites originais de nossas questões de pesquisa. Por exemplo, no decorrer de nossa pesquisa percebemos que o papel que um arquiteto de software, quando existe, irá desempenhar depende da empresa em que ele trabalha, local e experiência; não havendo, assim, atividades que sejam comuns a todos os arquitetos de software. Podemos observar na Figura 7a (segunda pergunta) que mais de 85% dos respondentes concordam com essa afirmação. Isto é possível ser visualizado, também, na resposta do entrevistado E14:

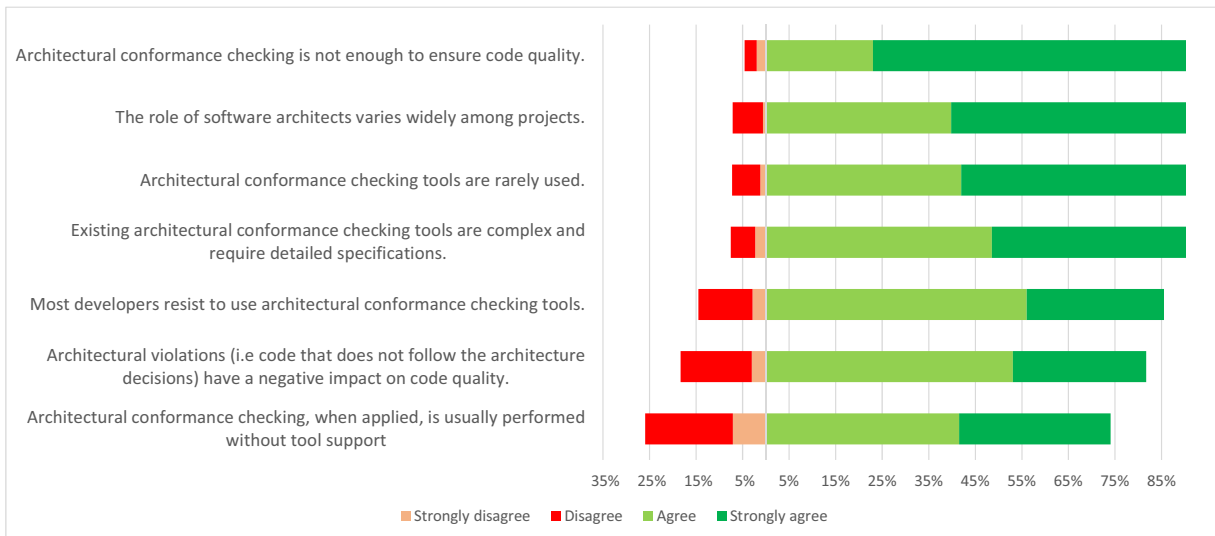
“[...] a profissão, o papel do arquiteto não é muito bem definido. E então isso não aumenta na responsabilidade dele. Não está certo e definido o que, quem tem que fazer isso ou que isso tem

que ser feito. Desde que foi definido arquiteturas, isso foi implementado de acordo com a arquitetura, e passou pela qualificação, tendo a correção de defeito e então, está tudo certo” (E14).

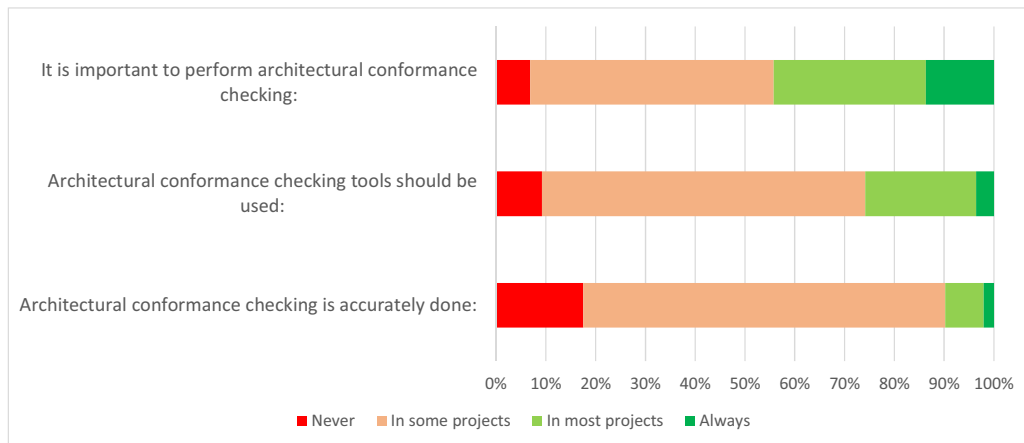
Levantamos, ainda, a reflexão sobre o quanto a indústria e a academia desempenham suas atividades de forma separada. Na Figura 5a (quarta pergunta) podemos observar que mais de 75% dos respondentes concordam que a academia e a indústria não demonstram interesse em sincronizar suas agendas. Além dessa confirmação, podemos refletir mais sobre esse ponto observando as respostas dos entrevistados E10 e E12:

“(…) pra mim é inútil eu pensar em uma técnica arquitetural, do ponto de vista acadêmico, ou pensar numa prática arquitetural, ou numa ferramenta, se eu não submeto essa ferramenta, dentro de um projeto real (...) as pesquisas em arquitetura elas têm que buscar conexão dos centros de pesquisa nas universidades com projetos do mundo real, ou seja, medir projetos, acompanhar, aplicar as ferramentas, pegar feedback, realmente pra que a pesquisa seja útil.” (E10)

“What I found out is that all the stuff that you learn in the computer engineering classes at the university you can forget about it every day you got to work in a real company. Because you learn a lot of things like: UML and component modeling standards. And you learn a lot of formal ways for describing software systems architecture. I were aware when you go to the reality these formal solutions are not used at all (...) So one important thing is that there is a huge disconnect between theory and practice” (E12)



a: Escala Likert.



b: Escala de frequência.

Figura 7: Afirmações sobre verificação de conformidade arquitetural e suas ferramentas (RQ3 e RQ4).

IV. TRABALHOS RELACIONADOS

Nesta seção, reunimos estudos relacionados a percepção de desenvolvedores através de surveys. Singer et al. [5] focou no entendimento sobre a relação entre os desenvolvedores e o Twitter. Os autores decidiram pesquisar sobre aspectos relacionados aos desenvolvedores que utilizam o Twitter em seus trabalhos. O estudo foi baseado na metodologia de Teoria Fundamentada (*Grounded Theory*) e foi dividido em 3 etapas: survey exploratório, entrevista e survey de validação. Os surveys foram enviados para usuários do GitHub que possuíam e-mail público. Nosso estudo não possui nenhuma relação com os resultados adquiridos por Singer et al.; porém, nos inspiramos nesse paper para planejar nossa metodologia de pesquisa.

Malavolta et al. [14] focaram na percepção de profissio-

nais com relação a linguagens de definição arquitetural, e.g., UML. O objetivo do estudo foi entender os pontos fortes, as limitações e as necessidades relacionadas às linguagens arquiteturais. Participaram, deste estudo, 48 profissionais de 40 empresas de TI em 15 países diferentes. Como resultados principais do estudo, os autores observaram que as linguagens arquiteturais acadêmicas não suprem as principais necessidades da indústria. Diferentemente dos profissionais da indústria, os profissionais da academia compartilham a crença da necessidade de linguagem formal e de domínio específico. Dentre as funcionalidades úteis para a indústria estão sintaxe gráfica, ferramentas de suporte e suporte para múltiplas visões da arquitetura. Enquanto isso, a engenharia reversa está entre as funções não úteis para a indústria.

Por fim, os autores perceberam que entender o que os

arquitetos de software precisam pode ajudar a direcionar os pesquisadores da área e que as linguagens normalmente utilizadas na indústria são as construídas por ela mesma, i.e. não as oriundas da academia. As conclusões apresentadas por Malavolta et al. dão direcionamentos para preencher a lacuna que existe entre academia e indústria. Além da área e de ser um estudo qualitativo, este trabalho se relaciona com o nosso pelo fato de pesquisar sobre lacunas entre a academia e a indústria, mostrando que não é apenas uma percepção e conclusão particular. Nosso trabalho vai além de linguagens arquiteturas, como é focado por Malavolta et al. Nosso objetivo foi ter uma visão mais ampla sobre a área, envolvendo os principais pontos: definição, documentação, verificação e ferramentas.

Heesch et al. [20] procuraram entender, a partir de um survey, como os arquitetos de software realizam suas tomadas de decisões, como priorizam os problemas, análises e soluções para a arquitetura. Este trabalho se relaciona com o nosso por procurar arquitetos profissionais para entender uma parte do contexto arquitetural na prática. Sua preocupação, semelhante à nossa, é que a academia possuía um baixo conhecimento sobre a tomada de decisões e, por esse motivo, não há ferramentas para auxiliar esta atividade. Assim como nosso estudo, eles perceberam que documentação arquitetural é uma atividade importante, que é necessário compreender os requisitos que envolvem o sistema e o nível de descrição da arquitetura depende de orçamento e tempo do projeto.

É importante notar que nosso estudo aborda aspectos gerais de arquitetura de software, abrangendo desde a definição de arquitetura a ferramentas de verificação de conformidade. No melhor do nosso conhecimento, os estudos realizados por Malavolta et al. e Heesch et al. são os mais relacionados à nossa pesquisa.

V. AMEAÇAS À VALIDADE

Em nosso estudo, temos a ameaça externa de não poder generalizar os resultados, uma vez que nossa amostra é pequena com relação a todos os profissionais da área. Para contornar esta situação, tivemos a preocupação de selecionar profissionais de diferentes linguagens, lugares, tempo de experiência e de diferentes tamanhos de empresas.

Também temos a ameaça interna de seleção de pessoas candidatas a responder nossas etapas. Consideramos que nossos critérios utilizados nas 3 (três) etapas foram suficientes para permitir uma variabilidade de profissionais respondentes, diminuindo, assim, essa ameaça.

Por fim, temos a ameaça de constructo, uma vez que nossos questionários e roteiros, apesar de terem sido planejados e revisados, podem ter sido construídos de forma que não tenha capturado toda a riqueza do nosso objeto de estudo. Além disso, nossos respondentes podem ter tentado responder a partir do que eles consideravam corretos para nossa pesquisa. Com relação a isso, tentamos ser o mais imparcial possível e revisamos todas as conclusões e análises. Entretanto, nossas análises e conclusões podem ter sofrido um viés com a visão de mundo do pesquisador, uma vez que temos nossos pré-conhecimentos do assunto.

VI. CONCLUSÕES

Neste artigo, apresentamos um estudo empírico para compreender melhor as percepções sobre Arquitetura de Software dentro da indústria. Compreendendo melhor o contexto prático sobre arquitetura de software, a academia pode voltar suas pesquisas para questões que sejam mais utilizadas nos processos de desenvolvimento de sistemas reais.

Percebemos que não existe uma única definição para arquitetura de software, podendo esta ser constituída por diversos aspectos do sistema. A academia considera a documentação arquitetural como uma atividade essencial no desenvolvimento de sistemas, trazendo diversas vantagens como o alinhamento do conhecimento entre todos os membros da equipe, por exemplo. No nosso estudo, os profissionais concordam que a documentação arquitetural é uma atividade importante. Contudo, esta atividade depende do escopo de cada projeto, da cultura da empresa e dos profissionais que trabalham nela.

Com relação à verificação de conformidade arquitetural e suas ferramentas, os profissionais mostraram: não conhecer muitas ferramentas de apoio à esta atividade; não ter muito interesse em conhecê-las; confiar mais na forma manual de verificação; considerá-las complicadas de utilizar e/ou não possuírem documentação arquitetural suficiente para utilizá-las. Enquanto isso, a academia continua pesquisando sobre verificação de conformidade arquitetural e apresentando novas ferramentas de apoio. Consideramos que academia e indústria devem caminhar juntas para evoluir e aperfeiçoar os processos de desenvolvimento de sistemas reais. A academia, em nossa visão, poderia olhar mais para as necessidades da indústria e abrir seu leque de conceitos arquiteturas para realizar pesquisas e ferramentas que possam ser utilizados na indústria. Por sua vez, a indústria poderia dialogar mais com os pesquisadores da academia, buscando um trabalho mais colaborativo, visando a eficiência e qualidade de seus produtos.

Nosso estudo confirmou estudos anteriores, afirmando que existe uma lacuna entre a academia e a indústria. Ambos não possuem uma única definição arquitetural. Com isso, cada ambiente realiza estudos e desenvolve os sistemas da sua própria maneira, de acordo com sua experiência e os fatores que afetam suas escolhas. Os resultados do nosso estudo são reforçados por 395 profissionais da indústria, onde 58 participaram do survey exploratório (com um subconjunto participando das entrevistas) e 337 participaram do survey confirmatório.

REFERÊNCIAS

- [1] G. Santos, N. Anquetil, A. Etien, S. Ducasse, M.T. Valente. *OrionPlanning: Improving Modularization and Checking Consistency on Software Architecture*. VISSOFT 2015.
- [2] V. Braun and V. Clarke. *Using Thematic Analysis in Psychology*. *Qualitative Research in Psychology*, pages 77–101, 2006.
- [3] V. Braun and V. Clarke. *Teaching Thematic Analysis: Overcoming Challenges and Developing Strategies for Effective Learning*. *The Psychologist*, pages 120–123, 2013.
- [4] L. Cohen, L. Manion, and K. Morrison. *Research Methods in Education*. Routledge, sixth edition, 2007.
- [5] L. Singer, F. Figueira Filho, and M. Storey. *Software Engineering at the Speed of Light: How Developers Stay Current Using Twitter*. In 36th International Conference on Software Engineering, 2014.

- [6] J. Knodel, D. Muthig, M. Naab, and Mikael Lindvall. *Static Evaluation of Software Architectures*. In 10th European Conference on Software Maintenance and Reengineering, pages 279–294, 2006.
- [7] J. Knodel and D. Popescu. *A Comparison of Static Architecture Compliance Checking Approaches*. In IEEE/IFIP Working Conference on Software Architecture, pages 44–53, 2007.
- [8] I. Melo, J. Brunet, D. Guerrero, and J. Figueiredo. *Verificação de Conformidade Arquitetural com Testes de Design – Um Estudo de Caso*. I Workshop Brasileiro de Visualização, Evolução e Manutenção de Software (VEM – CBSof), 2013.
- [9] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, M. Little, P. Merson, R. Nord, and J. Stafford. *Documenting Software Architectures: Views and Beyond*. AddisonWesley Professional, second edition, 2002.
- [10] H.P. Breivold, I. Crnkovic, and M. Larsson. *A Systematic Review of Software Architecture Evolution Research*. Information and Software Technology archive, 54:16–40, 2012.
- [11] D.E. Perry and A.L. Wolf. *Foundations for the Study of Software Architecture*. Software Engineering Notes, 4th Edition, 1992.
- [12] J. Brunet, D. Guerrero, and J. Figueiredo. *Design Tests: An Approach to Programmatically Check you Code Against Design Rules*. In Proceedings of the 31st International Conference on Software Engineering (ICSE 2009), New Ideas and Emerging Results, May 2009.
- [13] M. Eichberg, S. Kloppenburg, K. Klose, and M. Mezini. *Defining and Continuous Checking of Structural Program Dependencies*. In 30th International Conference on Software Engineering, pages 391–400, 2008.
- [14] I. Malavolta, P. Lago, H. Muccini, P. Pelliccione, and A. Tang. *What Industry Needs from Architectural Languages: A Survey*. Software Engineering, IEEE Transactions on (Volume:39 , Issue: 6), pages 869 – 891, 2013.
- [15] R. Terra, L. Passos, M.T. Valente, R. Diniz, and N. Mendonça. *Static Architecture Conformance Checking: An Illustrative Overview*. IEEE, 2010.
- [16] M. Shahin, P. Liang, and M.A. Babar. *A Systematic Review of Software Architecture Visualization Techniques*. Journal of Systems and Software, 94:161–185, 2014.
- [17] S. Ducasse and D. Pollet. *Software Architecture Reconstruction: a Process-Oriented Taxonomy*. Software Engineering, IEEE Transactions on, 35:573 – 591, 2009.
- [18] C.S.C. Rodrigues and C.M.L. Werner. *Uma Revisão Sistemática sobre as Iniciativas Realizadas no Ensino de Arquitetura de Software*. Rio de Janeiro: COPPE/UF RJ, ES-728/09, 2009.
- [19] M. Petre. *UML in Practice*. In 35th International Conference on Software Engineering, pages 722–731, 2013.
- [20] U. van Heesch and P. Avgeriou. *Mature Architecting - A Survey About the Reasoning Process of Professional Architects*. In Software Architecture (WICSA), 9th Working IEEE/IFIP Conference on, pages 260 – 269, 2011.
- [21] J. Brunet, G.C. Murphy, D. Serey, and J. Figueiredo. *Five Years of Software Architecture Checking: A Case Study of Eclipse*. IEEE Software, 2014.
- [22] R. Terra. *Conformação Arquitetural Utilizando Restrições de Dependência entre Módulos*. Dissertação de Mestrado. Belo Horizonte, Brasil, 2009.
- [23] <http://www.qsrinternational.com/product>. Último acesso em 29/07/2016