

A compreensão de reporte de *bugs* no desenvolvimento e uso de software: uma representação do conhecimento por meio de ontologia

RESUMO

[Mariana Oliveira

Nelson Tenório

Flávio Bortolozzi

O conhecimento vem sendo utilizado em diferentes organizações como fonte de lucro e vantagem competitiva. Nesse sentido, a Gestão do Conhecimento (GC) torna-se essencial para apoiar as organizações por meio da criação, armazenamento, recuperação e representação do conhecimento. Nas organizações que trabalham com o desenvolvimento de software, o conhecimento é essencial para que o produto final - o software - satisfaça a necessidade de seus clientes. Entretanto, o desenvolvimento desse sistema é uma tarefa complexa e que envolve conhecimento técnico apurado e um gerenciamento rigoroso por parte da organização, para que sejam respeitados os prazos e custos. Assim, é frequente a entrega de um software ao cliente que contém defeitos, que são comumente denominados de bugs. Quando o cliente identifica bugs no software, ele deve reportar ao fornecedor do produto, para que seja providenciada a sua correção, evitando assim maiores prejuízos. No entanto, o reporte de bug não é estruturado e nem suficiente para o fornecedor do software entender e solucionar o problema, fazendo-se necessárias diversas interações entre o cliente e o fornecedor. Portanto, o objetivo deste artigo é validar uma ontologia no sentido de aprimorar a compreensão de reportes de bugs durante o desenvolvimento e o uso de um software. Esta pesquisa caracteriza-se como de natureza aplicada com característica mista de procedimentos descritivo e exploratório. A pesquisa foi delineada em duas fases: A primeira fase da pesquisa consistiu no estudo qualitativo e quantitativo. A segunda fase, os resultados apresentam quatro ontologias modeladas para reportes de bug e a validação dessas ontologias por meio da sua extração em forma textual.

PALAVRAS-CHAVE: Extração. Bug Fixing. Qualidade.

INTRODUÇÃO

As organizações estão cada vez mais utilizando o conhecimento como fonte de inovação para garantir a sustentabilidade nos mercados em que atuam. Dessa forma, no final do século XX, ficou evidente que o conhecimento se tornou o principal fator de produção para as organizações, sendo o responsável por mudanças estruturais e produtivas (SANTOS et al., 2016). Nesse sentido, a Gestão do Conhecimento (GC) pode ser destacada para auxiliar na estruturação e sistematização do conhecimento.

Segundo Kamara, Anumba e Carrilho (2002) a GC tem como objetivo aperfeiçoar o conhecimento para que as organizações alcancem melhor desempenho, maior valor, competitividade, vantagem, e retorno sobre o investimento. No entanto, o conhecimento também pode ser aperfeiçoado por meio dos processos da GC, tais como, criação, armazenamento, recuperação e representação desse conhecimento. Desse modo, a representação do conhecimento é um processo da GC de grande relevância para as organizações. Para Brascher e Café (2009) a representação do conhecimento reflete um modelo de abstração do mundo real, construído para determinada finalidade. Além disso, a representação do conhecimento tem como finalidade aperfeiçoar a comunicação, o relacionamento social e auxiliar na recuperação de informações da organização (JOB, 2008).

Nas organizações da indústria de software o conhecimento também é essencial para o produto final, nesse caso, o próprio software. Nessas organizações, uma das premissas básicas é a geração e disseminação de conhecimento. Desse modo, a importância das atividades da GC é mais evidenciada nos fornecedores de software (GASPAR et al., 2016). Além disso, essas organizações se destacam por buscar melhorias contínuas em seus processos e atividades organizacionais e empenhar-se em utilizar o conhecimento organizacional como um recurso valioso para se destacar em um mercado caracterizado por muita concorrência e competitividade.

Diante disso, podem-se destacar algumas atividades desenvolvidas na construção do software, tais como: levantamento de requisitos, implementação, correção e testes. No caso específico das correções do software, uma atividade de grande relevância são os reportes de bug. Os bugs são defeitos ou falhas no software que indica um comportamento inesperado do sistema (GEGICK; ROTELLA; XIE, 2010). Dessa forma, quando um cliente encontra um bug durante o uso do software, ele deve ser reportado rapidamente ao fornecedor desse sistema para que seja providenciada a sua correção. Esses reportes apresentam informações relevantes para auxiliar na tarefa de correção dos bugs, também conhecida como *bug fixing*, que compreende em localizar e solucionar o bug reportado pelo cliente.

Nesse cenário, o objetivo deste artigo é validar uma ontologia no sentido de aprimorar a compreensão de reportes de bugs durante o desenvolvimento e o uso de um software. Dessa forma, este artigo está organizado em cinco seções. Além desta Introdução, a segunda seção trata da Gestão do Conhecimento na indústria de software, contendo uma subseção que discorre sobre ontologias e reportes de bug. Na terceira seção relatam-se os procedimentos metodológicos empregados na pesquisa, que clarifica como os dados foram obtidos e analisados, e na seção seguinte são apresentados os resultados e discussão desta pesquisa.

Por fim, na quinta seção constam as conclusões da pesquisa, seguida das referências utilizadas.

GESTÃO DO CONHECIMENTO NA INDÚSTRIA DE SOFTWARE

Na indústria de software, gerenciar o conhecimento é uma tarefa difícil, pois essa indústria faz uso intensivo do conhecimento e possui complexidade em suas tarefas. Um dos motivos dessa complexidade se dá pelo fato de que, independentemente do tamanho, realizam-se várias tarefas ao mesmo tempo (NAWINNA, 2011). Sendo assim, o conhecimento é utilizado constantemente por essas organizações pelo fato de produzirem um grande volume de informações constantemente (LOURENÇO et al., 2018) e também por apresentarem características específicas, devido ao fato de desenvolverem atividades que geram produtos de alto valor agregado (BJØRNSON; DINGSØYR, 2008).

No entanto, um dos principais desafios da indústria de software é de utilizar o conhecimento como uma forma de gerenciamento e problemas relacionados à gestão (SANTOS et al., 2018). Isso se faz importante, pois quando o conhecimento organizacional não é bem gerido pelas empresas de software, ele tende a ficar fragmentado (TENÓRIO et al., 2017). Para evitar tal fragmentação faz-se necessário adicionar o conhecimento a processos, assim é possível alcançar maior produtividade e inovação devido ao fato do conhecimento dos indivíduos estarem diretamente relacionados ao produto final (FENTON; BIEMAN, 2014).

Assim, a indústria de software, uma vez produtora e fornecedora de tecnologia está em um cenário de constantes mudanças que necessita de modelos e processos que possibilitem aos indivíduos renovar e aprimorar o seu conhecimento. Contudo, como a GC se caracteriza pelo planejamento, organização, motivação, controle de pessoas, processos e sistemas (KING, 2009), ela se torna essencial para as organizações da indústria de software buscarem melhores resultados no mercado. Além disso, saber gerenciar esse conhecimento é fundamental para que essas organizações inovem seus processos, atividades e consigam maiores lucro e produtividade em um ambiente competitivo.

Ontologias e Reportes de Bug

Uma ontologia é essencial para a recuperação, compartilhamento e representação do conhecimento dentro das organizações. Vasanthapriyan, Thian e Zhao (2017) definem uma ontologia como uma forma de estabelecer uma conceituação comum e coerente a fim de facilitar a gerência do conhecimento nas organizações. Além disso, para Estevão e Strauhs (2013) uma ontologia pode ser uma forma de melhorar a performance da recuperação da informação.

Gruber (1995) estabelece um conjunto de critérios para a modelagem de uma ontologia que devem considerar a clareza, consistência e vocabulário. Em relação à clareza, uma ontologia deve comunicar efetivamente os termos definidos e todas as definições devem ser documentadas para facilitar o entendimento dos usuários. Considerando a consistência, uma ontologia deve ter consistência interna e lógica. A consistência deve também aplicar às partes das definições, como a documentação. Finalmente, no que se refere ao vocabulário, uma ontologia deve ser projetada para antecipar o uso do vocabulário compartilhado, oferecendo uma base conceitual para uma série de tarefas

antecipadas. Nesse cenário, o uso de ontologias pode ser um apoio para relatar bugs em sistemas. Isso porque, em um relato de bug de um sistema apresentar uma descrição da ocorrência desses bugs durante o uso do software (BETTENBURG et al., 2010).

No entanto, não existe uma padronização na indústria de software para os reportes de bug, mas sim recomendações (KAUR; JINDAL, 2017; CAVALCANTI, 2009). Em geral, um bug é reportado com o preenchimento de algumas informações-padrão como versão do software, módulo afetado, hora da ocorrência do bug, dentre outras, além de ser descrito em linguagem natural, o que muitas vezes dificulta o entendimento do reporte do bug. Além disso, quando um grande número de reporte do bug é aberto, ele pode conter uma dezena de comentários e milhares de frases dificultando o rastreamento do conhecimento construído, o que consome uma quantidade substancial de tempo (FERREIRA et al., 2016).

Entretanto, o reporte do bug é preenchido de forma inapropriada, dificultando a sua identificação, reprodução e correção. Desse modo, os reportes de bug tornam-se incompletos e imprecisos não fornecendo todas as informações necessárias para a sua correção (BETTENBURG, 2010). Além disso, o autor salienta que os desenvolvedores gastam mais tempo com relatos de bug mal escritos, pois a identificação do problema nesses relatos dispendem mais tempo. Portanto, a qualidade de um reporte de bug é essencial para a sua correção. Segundo Hooimeijer e Weimer (2007) os reportes de bug de qualidade são tratados mais rapidamente do que os relatos incompletos. Bettenburg et al. (2010) apresentam algumas informações que devem constar nos reportes de bug e que são essenciais para a sua correção.

Portanto, as organizações interessadas na qualidade de reportes de bug, conseguem diminuir o tempo de correção do bug e diminuir o retrabalho, consequentemente, reduzindo os custos de desenvolvimento e manutenção do software. Nesse contexto, uma ontologia aprimora os reportes de bug facilitando a sua compreensão durante o desenvolvimento e a manutenção do software.

METODOLOGIA

Esta pesquisa possui natureza aplicada, com objetivo descritivo e exploratório. A abordagem desta pesquisa refere-se a de métodos mistos, com fonte de informação de campo por meio de procedimentos técnicos de questionários e entrevistas. Essa pesquisa foi elaborada em quatro etapas, sendo elas, dois questionários, modelagem e validação da ontologia. Dessa forma, o primeiro questionário elaborado para essa pesquisa foi adaptado de dois questionários de uma pesquisa de Bettenburg et al. (2010). Esse questionário possui como finalidade entender quais as principais dificuldades que os desenvolvedores e suportes enfrentam ao relatar um bug, e a partir disso iniciar a modelagem da ontologia. A diferença entre esses questionários é que para essa pesquisa os questionários foram aplicados para uma empresa privada, e os questionários de Bettenburg et al. (2010) foram aplicados em comunidades open source. A elaboração e administração do questionário em uma organização da indústria de software da região de Maringá, Paraná, Brasil. Nesse sentido, os cargos de suportes e desenvolvedores foram selecionados, pois são os que tratam diretamente com o relato de bug do cliente. Assim, selecionou-se 13

suportes e 13 desenvolvedores como respondentes. A coleta de dados dos questionários ocorreu nos meses de abril e maio de 2018.

A segunda etapa consistiu na elaboração e execução de um protocolo para realização de entrevistas semiestruturadas, explorando e aprofundando o conhecimento sobre relatos de bug e ontologias na indústria de software. Essa etapa auxiliou na modelagem da ontologia. Além disso, nessa etapa foi possível identificar os quatro principais relatos de bugs que os suportes e desenvolvedores tiveram maiores dificuldades de entendimento, esses relatos de bugs foram estudados e utilizados para auxiliar na modelagem das ontologias. Essas entrevistas foram realizadas com suportes e desenvolvedores de software. A coleta de dados das entrevistas ocorreu nos meses de fevereiro e março de 2018.

A terceira etapa refere-se a modelagem da ontologia. Assim, após as análises das metodologias de ontologias da literatura e dos dados coletados com o questionário da primeira etapa e com as entrevistas foi possível identificar algumas etapas para a modelagem das ontologias de relatos de bug. A metodologia para a modelagem da ontologia é uma técnica necessária, que busca facilitar e criar um padrão para alcançar um objetivo desejado. Essa pesquisa utilizou-se como base algumas fases da modelagem de Fernández et al., (1997). Além disso, essa metodologia foi a que mais se identificou com o propósito das ontologias modeladas. Assim, as principais fases utilizadas para a modelagem da ontologia foram o planejamento, especificação, aquisição do conhecimento, conceitualização e validação da ontologia. A fase de planejamento tem como objetivo identificar as principais tarefas da ontologia, e planejar a utilização dos recursos dessa ontologia. Nessa fase, foi utilizada um instrumento de pesquisa para entender as principais características de um reporte de bug de um software privado. A fase de especificação define o problema que a ontologia precisa resolver e quem serão os usuários da ontologia. Essa fase utilizou-se de entrevistas com profissionais da área (suportes e desenvolvedores) para entender quais os principais problemas que eles enfrentam no entendimento de relato de bug. Na fase de aquisição do conhecimento é possível adquirir conhecimento sobre a ontologia. Nessa fase, foram utilizadas entrevistas com profissionais da área que a ontologia está sendo modelada. Na conceitualização é a modelagem conceitual da ontologia que apresenta o problema e a solução por meio da ontologia. Essa fase foi modelada a ontologia por meio do software protege. Após a modelagem, foi possível montar um texto para cada ontologia. Esse texto foi montado de acordo com o passo a passo que a ontologia forneceu na modelagem. Por fim, a validação tem como objetivo verificar se a ontologia modelada foi relevante para a pesquisa. Nessa fase foi utilizada um instrumento de pesquisa no intuito de validar o texto do reporte de bug real reportado pelo cliente e o texto da extração da ontologia.

Após a modelagem da ontologia, foi aplicado um questionário com 16 fornecedores de software para a validação da ontologia. A coleta de dados ocorreu no mês de novembro de 2019. E a seleção dos participantes aconteceu com uma amostragem intencional (CRESWELL; CLARK, 2013). Além disso, foram selecionados 16 respondentes de 4 diferentes empresas para a validação da ontologia. O questionário foi dividido em quatro módulos, sendo eles, tributário, recursos humanos, administrativo e contábil. Dessa forma, o questionário consistiu em dois relatos de bug para cada ontologia modelada. O primeiro relato de bug apresentado no questionário consistia em um relato real reportado pelo

cliente de uma empresa fornecedora de software, e o segundo relato consistia na estruturação de um texto por meio da ontologia. Dessa maneira, o relato de bug real reportado foi comparado com o texto desenvolvido por meio da ontologia.

RESULTADOS E DISCUSSÕES

Principais itens de um relato de Bug

No desenvolvimento de software, os relatos de bugs devem fornecer informações relevantes para que os desenvolvedores façam a correção. No entanto, na maioria das vezes esses relatos não apresentam todas as informações necessárias para fazer a correção em um tempo hábil. Assim, esse instrumento de pesquisa apresentou dados de uma empresa de software privado, e é feito uma comparação a pesquisa de Bettenburg (2010) que apresenta dados de um software livre. Nesse sentido, foi necessário um instrumento para identificar quais as informações necessárias que um relato de

bug deve apresentar para tornar a correção do bug mais facilitada. A Tabela 1 apresenta todos os itens indicados pelos desenvolvedores para corrigir bugs.

Tabela 1 – Conteúdos dos relatórios de Bugs mais utilizados pelos desenvolvedores

ESTUDO QUANTITATIVO	BETTENBURG et al., (2010)
Capturas de tela (76,90%)	Capturas de tela (75%)
Relatórios de erros (69,20%)	Relatórios de erros (65%)
Passos para reproduzir o bug (61,50%)	Passos para reproduzir o bug (97%)
Comportamento do sistema (61,50%)	Comportamento do sistema (95%)
Informações sobre os casos de testes (53,80%)	Informações sobre os casos de testes (85%)
Comportamento esperado do sistema (53,80%)	Comportamento esperado do sistema (89%)
Nome do produto (46,20%)	Nome do produto (65%)
Informações (46,20%)	Informações (62%)
Gravidade (38,50%)	Gravidade (47%)
Versão do sistema (23,10%)	Versão do sistema (75%)
Informações sobre o sistema operacional (23,10%)	Informações sobre o sistema operacional (63%)
Resumo (23,10%)	Resumo (81%)
Nome dos componentes (15,40%)	Nome dos componentes (67%)
Hardware (15,40%)	Hardware (32%)
Exemplo de códigos (15,40%)	Exemplo de códigos (68%)
Rastreamento de pilha (7,70%)	Rastreamento de pilha (89%)

Fonte: Dados da pesquisa (2019).

Comparando-se os dados obtidos nesta pesquisa (Tabela 1) com os dados obtidos por Bettenburg et al. (2010), verificam-se poucas convergências entre os cinco itens mais indicados em cada pesquisa, uma vez que, na pesquisa do Bettenburg et al. (2010), Passos para reproduzir o bug e Comportamento do Sistema também estão entre os cinco itens mais indicados pelos desenvolvedores, com participações, respectivamente, de 97%, 95%. Já o item rastreamento de pilha obteve 89%, sendo o quarto item mais indicado por desenvolvedores no trabalho de Bettenburg (2010). Já nessa pesquisa, rastreamento de pilha obteve somente 7,70%.

Já os suportes que participaram desta pesquisa, relataram os principais itens ao preencher os relatos de bug. A Tabela 2 apresenta todos os itens relatados pelos suportes ao preencher relatórios de bugs.

Tabela 2 - Conteúdos dos relatórios de Bugs mais relatados pelos suportes

ESTUDO QUANTITATIVO	BETTENBURG et al., (2010)
Nome do produto (76,90 %)	Nome do produto (94%)
Gravidade (69,20%)	Gravidade (77%)
Capturas de tela (69,20%)	Capturas de tela (60%)

Passos para reproduzir o bug (61,50%)	Passos para reproduzir o bug (98%)
Informações sobre os casos de testes (53,80%)	Informações sobre os casos de testes (56%)
Exemplo de códigos (46,20%)	Exemplo de códigos (36%)
Relatórios de erros (46,20%)	Relatórios de erros (53%)
Versão do sistema (38,50%)	Versão do sistema (91%)
Resumo (38,50%)	Resumo (90%)
Comportamento do sistema (30,80%)	Comportamento do sistema (96%)
Comportamento esperado do sistema (30,80%)	Comportamento esperado do sistema (89%)
Informações sobre o sistema operacional (7,70%)	Informações sobre o sistema operacional (90%)
Rastreamento de pilha (7,70%)	Rastreamento de pilha (50%)
Nome dos componentes (0%)	Nome dos componentes (87%)
Hardware (0%)	Hardware (48%)
Informações (0%)	Informações (60%)

Fonte: Dados da pesquisa (2019).

Comparando-se os dados obtidos nesta pesquisa (Tabela 2) com os dados obtidos por Bettenburg et al. (2010), verificam-se muitas divergências. Entre os cinco itens mais indicados em cada pesquisa, apenas dois itens se assemelham: Nome do Produto e Passos para reproduzir o bug. Na pesquisa do Bettenburg et al. (2010) esses itens tiveram participação de 94% e 98%, respectivamente. Nessa pesquisa, esses itens obtiveram 76,90% e 61,50%, respectivamente. Já entre os cinco itens menos citados em ambas as pesquisas, apenas Hardware e Rastreamento de Pilha são itens em comum. Na pesquisa do Bettenburg et al. (2010) esses itens tiveram participação de 48% e 50%, respectivamente. Já nessa pesquisa, esses itens foram obtiveram 0% e 7,70%. Dessa forma, foi possível verificar com os dados coletados nesta pesquisa que há certa discrepância entre os itens utilizados pelos desenvolvedores para corrigir os bugs indicados pelos suportes, como relatados ao preencher relatórios de bugs. Portanto, fica evidente que os desenvolvedores e suportes nessa pesquisa apresentam respostas diferentes, reforçando a ideia de que uma ontologia é essencial para os membros de uma equipe ter um vocabulário comum entre eles. Além disso, os principais itens descritos como importantes em um relato de bug por desenvolvedores e suportes foram relevantes para insights da modelagem da ontologia.

Resultados do protocolo de entrevistas semiestruturadas

Os resultados das entrevistas mostraram que os fornecedores de software enfrentam grandes dificuldades no entendimento de um relato de bug. Dessa forma, quando questionados sobre quais as dificuldades que enfrentam ao ler um relato de bug um dos entrevistados revelou que:

“Tem relatos de bug que são explícitos, ali na cara e a gente já corrige, daí beleza, mas tem relatos que às vezes a gente nem consegue ter o entendimento para fazer a correção do bug” (DESENVOLVEDOR I, 11/02/2019).

Além disso, os resultados apontaram que os fornecedores de software apresentam dificuldades específicas no entendimento de um relato de bug. Essas falas corroboram o exposto por Chaparro (2017) em que os fornecedores de software confiam nas informações textuais essenciais dos relatos de bugs para triar e corrigir erros de software. Infelizmente, embora relevantes e úteis, essas informações geralmente estão ausentes, incompletas, superficiais, ambíguas ou complexas, dificultando o entendimento do bug. Para Joorabchi, Mirzaaghaei e Mesbah (2014), os reportes de bugs têm conteúdo textual de baixa qualidade. Assim, quando questionados quais os tipos de dificuldades que eles enfrentavam ao interpretar um relato de bug um dos entrevistados salientou que:

“Até conseguirmos chegar aonde ocorreu o bug, mas às vezes é bem mal escrito e difícil de entender o relato, então a gente tem que chamar o cara, conversar, às vezes só com o texto do relato não resolve”. (DESENVOLVEDOR II, 04/02/2019).

Dessa forma, fica evidente que os fornecedores de software apresentam dificuldades no entendimento, principalmente na escrita de um relato de bug. Para Joorabchi, Mirzaaghaei e Mesbah (2014) conteúdo de baixa qualidade é uma das principais razões para falta de entendimento de relatos de bugs não reproduzidos, pois os fornecedores de software precisam gastar mais tempo e esforço para entender as descrições ou pedir para esclarecimentos e informações adicionais de um relato de bug. Um dos entrevistados salientou que:

“A descrição do relato de bug é falha às vezes, isso porque faltam informações ou até mesmo essas informações não são claras o suficiente. Assim, geram-se dúvidas, compreensões erradas a respeito do problema, sendo então, questões que dificultam o entendimento”. (SUPORTE II, 18/02/2019).

Para Bettenburg (2010) os fornecedores de software apresentam dificuldades de entendimento nos relatos mal escritos, e conseqüentemente, desperdiçam muito tempo para encontrar o bug. Além disso, os entrevistados foram questionados quais tipos de ações facilitariam para o entendimento de um relato de bug. Assim, um dos entrevistados evidenciou que:

“Acho que se o cliente conseguisse relatar de forma mais clara, se a gente fosse mais fechado, traçasse um caminho para ele relatar esse bug, ficaria uma maneira mais tranquila de entendimento.” (DESENVOLVEDOR II, 18/02/2019).

Diante disso, fica evidente que os fornecedores de software apresentam uma grande dificuldade para entendimento de relatos de bug. Além disso, os fornecedores dispõem uma grande parte do tempo para encontrar onde está o bug para fazer a correção. Nesse sentido, uma ontologia para que aprimore o entendimento dos reportes de bug realizados por clientes do software torna-se essencial, pois a ontologia pode ser utilizada para compartilhar uma linguagem comum entre os membros de uma organização. Para Tim, Hendler e Lassila (2001) uma ontologia auxilia as organizações na estruturação das informações, auxiliando em tarefas consideradas **trabalhosas** com o objetivo de recuperar a

informação desejada. Além disso, os autores salientam que é fundamental o uso de ontologias, que fornecem um vocabulário compartilhado e comum para modelar determinado domínio (área do conhecimento), seus conceitos, propriedades e relações. Dessa maneira, a ontologia pode eliminar contradições e inconsistências, tornando a escrita mais uniforme, facilitando a compreensão de um relato de bug.

Modelagem da Ontologia

Para a modelagem da ontologia, por meio das entrevistas com os fornecedores de software, foi possível encontrar um conjunto de relatos de bugs que esses fornecedores tiveram mais dificuldade de entendimento. Assim, foram selecionados quatro relatos para a modelagem da ontologia, um para cada módulo, sendo eles, módulo administrativo, tributário, recursos humanos e contábil. A ontologia foi modelada de acordo com cada módulo, pois o sistema desenvolvido pela empresa é dividido de acordo com esses quatro módulos. Em relação ao primeiro módulo, o administrativo no qual diz respeito à problemas de entendimento com a solicitação de Melhorias na tela requisição de compra. Esse bug ocorreu em uma tela específica de requisição de compras. Essa tela é utilizada para solicitar compras para um setor específico responsável. Desse modo, o sistema está fornecendo um contrato de compra se for pesquisado pelo nome do representante do fornecedor. Já quando o usuário pesquisa pelo número do contrato, o sistema não apresenta nenhuma informação. Dessa forma, a pesquisa pelo número do contrato precisa de alteração.

Quadro 1 – Relato de bug original

RELATO DE BUG ORIGINAL
Solicito que seja arrumado na tela de requisição de compra, no campo contrato, hoje informo que hoje a autocomplete só está trazendo o contrato se pesquisar pelo nome do representante do fornecedor em outros casos até aparece o contrato se pesquisar pelo número, porém em outros não. por isso é necessário a correção desta funcionalidade.

Fonte: Autores (2020).

Nesse sentido, a partir do relato do cliente fez-se possível estruturar o seguinte texto a partir da ontologia:

Quadro 2 – Relato de bug estruturado pela ontologia

RELATO DE BUG ESTRUTURADO PELA ONTOLOGIA
Foi encontrado um bug no módulo administrativo no submodelo compras. Na tela de requisição de compra, está ocorrendo um erro ao realizar uma pesquisa pelo número de contrato. Favor corrigir a pesquisa pelo número de contrato.

Fonte: Autores (2020).

Em relação ao módulo recursos humanos que refere-se à erro na Sefip - Geração do arquivo faltando pessoas com duplo vínculo. Esse bug ocorreu devido ao fato de o relatório gerado não contemplar pessoas com duplos vínculos na empresa. Esse duplo vínculo é quando uma pessoa pode ter mais de um vínculo empregatício em uma mesma empresa/instituição.

Quadro 3 – Relato de bug original

RELATO DE BUG ORIGINAL
Em nova aliança do ivai foi verificado que o sefip não está gerando pessoa com duplos vinculos assim impossibilitando envio e informações corretas.

F

o

Fonte: Autores (2020).

Quadro 4 – Relato de bug estruturado pela ontologia

RELATO DE BUG ESTRUTURADO PELA ONTOLOGIA
Foi encontrado um bug no módulo recursos humanos e no submodelo administração de pagamentos na Sefip. Os dados estão inconsistentes devido a um erro na geração de pessoas de duplo vínculo. Favor corrigir a geração de pessoas de duplo vínculo.

Fonte: Autores (2020).

Em relação ao módulo contábil, em que o terceiro reporte de bug que os fornecedores de software tiveram problemas de entendimento foi o denominado de Contabilidade - Receita Realizada – Conta Inter Município. Esse bug foi causado em um sistema desenvolvido pela empresa no módulo contábil. Esse bug consiste em um erro no relatório de créditos a receber do sistema. Esse relatório fornece todos os créditos a receber que os clientes possuíam direito. Nesse sentido, os cálculos totais dos créditos dos clientes não estavam sendo apresentados.

Quadro 5 – Relato de bug original

RELATO DE BUG ORIGINAL
Corrigir a tela de receita realizada na receita de créditos de clientes.

Fonte: Autores (2020).

Assim, para facilitar o entendimento, foi possível extrair o seguinte texto da ontologia.

Quadro 6 – Relato de bug estruturado pela ontologia

RELATO DE BUG ESTRUTURADO PELA ONTOLOGIA
Foi encontrado um bug no módulo recursos humanos e no submodelo administração de pagamentos na Sefip. Os dados estão inconsistentes devido a um erro na geração de pessoas de duplo vínculo. Favor corrigir a geração de pessoas de duplo vínculo.

Fonte: Autores (2020).

O último reporte de bug que os fornecedores de software tiveram problemas de entendimento foi o denominado de Erro na geração de *QR Code* da certidão. Esse bug foi causado em um sistema desenvolvido pela empresa no módulo tributário. Essa certidão é referente a uma certidão negativa de débitos que é um documento que atesta a ausência de pendências de uma empresa ou indivíduo. Nessa certidão, quando o usuário fazia a impressão da certidão, o QR Code não aparecia para a impressão.

Quadro 7– Relato de bug original

RELATO DE BUG ORIGINAL
A rotina de geração e emissão de certidão está apresentando erro ao gerar certidões com QRCode. O documento é gerado normalmente, mas ao clicar no botão de imprimir, a certidão é exibida para o usuário sem o QRCode.

Fonte: Autores (2020).

Assim, para facilitar o entendimento, foi possível extrair o seguinte texto da ontologia.

Quadro 8 – Relato de bug estruturado pela ontologia

RELATO DE BUG ESTRUTURADO PELA ONTOLOGIA
Foi encontrado um bug no módulo Tributário e no submódulo Certidões. Na certidão negativa de débitos está ocorrendo um erro na geração do QR Code quando o usuário faz a impressão dessa certidão. Favor corrigir a geração do QR Code na certidão negativa de débitos quando é feita impressão.

Fonte: Autores (2020).

Validação da Ontologia

Ressalta-se que o objetivo desta pesquisa é validar uma ontologia que aprimore o entendimento dos reportes de bug, realizados por clientes do software, para a execução da tarefa de correção desses bugs, uma vez que essa validação tem como objetivo contribuir no entendimento de relatos de bug na indústria de software, fornecendo aos desenvolvedores e suportes uma ontologia para facilitar esse processo de entendimento. O instrumento proposto para a

validação da ontologia consiste em um questionário no qual apresenta como características a comparação entre um texto do reporte de bug real reportado por um cliente e um texto estruturado pela ontologia.

Na ontologia do módulo recursos humanos, o grau de entendimento dos respondentes em relação ao relato de bug real foi de 5 e o texto desenvolvido pela ontologia obteve grau de entendimento de 8,87. Além disso, (31,25%) dos respondentes tiveram entendimento máximo no texto estruturado pela ontologia. E nenhum dos respondentes mostraram um grau de entendimento para o texto estruturado pela ontologia inferior a 8. Quando questionados sobre qual relato de bug obtiveram um maior entendimento, (100%) dos respondentes salientaram que o texto estruturado pela ontologia é mais fácil de entender.

Na ontologia do módulo tributário o relato de bug real reportado foi comparado com o texto desenvolvido por meio da ontologia. Diante disso, o grau de entendimento apresentado no relato de bug real foi de 7,12 e o texto desenvolvido pela ontologia obteve grau de entendimento de 8,85. Além disso, (37,5%) dos respondentes tiveram um entendimento máximo do texto estruturado por meio da ontologia. Quando questionados sobre qual relato de bug obtiveram um maior entendimento, (81,25%) entenderam o texto estruturado pela ontologia e (18,75%) salientaram que o relato de bug real tinha um entendimento maior. Além disso, somente (12,5%) evidenciaram um grau de entendimento inferior a 8 para o texto estruturado pela ontologia. Dessa forma, fica evidente que a ontologia do módulo tributário também aprimora o entendimento dos relatos de bug. Na ontologia do módulo Administrativo o relato de bug real reportado também foi comparado com o texto desenvolvido por meio da ontologia. Assim, o grau de entendimento dos resultados apresentados no relato de bug real foi de 5,93 e o texto desenvolvido pela ontologia obteve grau de entendimento de 8,56. Além disso, (31,25%) dos respondentes obtiveram entendimento máximo no texto estruturado pela ontologia. Quando questionados sobre qual relato de bug obtiveram um entendimento maior, (87,5%) salientaram que o relato de bug estruturado pela ontologia é mais fácil de entender e somente (12,5%) salientaram que o texto real reportado é mais fácil de entender. Na ontologia do módulo contábil o relato de bug real reportado também foi comparado com o texto desenvolvido por meio da ontologia. Dessa forma, o grau de entendimento dos resultados apresentados no relato de bug real foi de 3,12 e o texto desenvolvido pela ontologia obteve grau de entendimento de 8,75. É possível evidenciar uma grande diferença entre os dois relatos dessa ontologia, o relato real reportado obteve um grau de entendimento muito abaixo quando comparado com o relato de bug estruturado pela ontologia. Além disso, (25%) dos entrevistados obtiveram entendimento mínimo do relato de bug real reportado, e (37,5%) obtiveram entendimento máximo no relato de bug estruturado pela ontologia. E (100%) dos respondentes entenderam mais facilmente o texto do relato de bug extraído pela ontologia. Dessa forma, após as análises dos resultados do instrumento de validação, foi possível destacar que a ontologia também pode ser uma poderosa forma de aprimorar o entendimento reportes de bug de fornecedores de software em organizações de desenvolvimento de software.

CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo validar as ontologias que aprimorem o entendimento dos reportes de bug, realizados por clientes do software, para a execução da tarefa de correção desses bugs. E solucionou a seguinte questão de pesquisa: “As ontologias modeladas podem aprimorar o entendimento dos relatos de bug por parte dos desenvolvedores para a execução da tarefa de correção desses bugs?”. Dessa forma, essas ontologias foram modeladas e validadas e os resultados mostraram que todos os respondentes da ontologia do módulo recursos humanos e contábil obtiveram um entendimento de (100%) no relato de bug reportado pela ontologia. Nesse sentido, as ontologias modeladas apresentaram um grau de entendimento superior a 8. Dessa forma, fica evidente que as ontologias podem aprimorar o entendimento de relatos de bugs de fornecedores de software. Em relações as contribuições da pesquisa, esse estudo é de grande relevância uma vez que contribui para que o setor de TI torna-se mais consolidado, proporcionando uma maior inovação, lucro e vantagem competitiva em um mercado altamente tecnológico. Além disso, esse estudo pode ser viável no sentido de diminuir o tempo e custo em relação à correção de relatos de bug. Contudo, como recomendações para pesquisas futuras, sugere-se estudar as melhores técnicas de implementação das ontologias em uma empresa de desenvolvimento de software e após o estudo dessas técnicas sugere-se a implementação da ontologia.

REFERÊNCIAS

- BETTENBURG, N. et al. What makes a good bug report? IEEE Transactions on Software Engineering, v. 36, n. 5, p. 618–643, 2010.
- BJØRNSON, F. O.; DINGSØYR, T. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. Information and Software Technology, v. 50, n. 11, p. 1055–1068, out. 2008.
- BRASCHER, M.; CAFÉ, L. Organização da informação ou organização do conhecimento? ENANCIB. Anais...Salvador: ENANCIB-USP, 2009
- CHAPARRO, O. Improving bug reporting, duplicate detection, and localization. Proceedings - 2017 IEEE/ACM 39th International Conference on Software Engineering Companion, ICSE-C 2017, p. 421–424, 2017.
- CLARK, V. L. P.; CRESWELL, J. W. Understanding research: A consumer’s guide. London: Pearson, 2015.
- CRESWELL, J. W. Research design : qualitative, quantitative, and mixed methods approaches. Thousand Oaks: Sage publications, 2013.

ESTEVIÃO, J. S. B.; STRAUHS, F. DO R. Proposta de uma ontologia como modelo de referência no domínio da Memória Organizacional Histórica. *Perspectivas em Ciência da Informação*, 2013.

FENTON, N.; BIEMAN, J. *Software metrics: a rigorous and practical approach*. 2. ed. London: International Thomson Computer Press, 2014.

GASPAR, M. A. et al. Gestão do Conhecimento em empresas atuantes na indústria de software no Brasil: um estudo das práticas e ferramentas utilizadas. *Informação e Sociedade*, v. 26, n. 1, p. 151–166, 2016.

GEGICK, M.; ROTELLA, P.; XIE, T. Identifying security bug reports via text mining: An industrial case study. *Proceedings - International Conference on Software Engineering*. Anais...2010

GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human - Computer Studies*, 1995.

JOB, I. Estudos Cognitivos E a Representação Do Conhecimento Na Ciência da Informação. *Revista ACB: Biblioteconomia em Santa Catarina*, v. 13, n. 2, p. 365–378, 2008.

JOORABCHI, M. E.; MIRZAAGHAEI, M.; MESBAH, A. Works for me! characterizing non-reproducible bug reports. 11th Working Conference on Mining Software Repositories, MSR 2014 - Proceedings. Anais...2014

KALLIO, H. et al. Systematic methodological review: developing a framework for a qualitative semi-structured interview guide *Journal of Advanced Nursing*, 2016.

KAMARA, J. M.; ANUMBA, C. J.; CARRILLO, P. M. A CLEVER approach to selecting a knowledge management strategy. *International Journal of Project Management*, 2002.

KING, W. Knowledge management and organizational learning. In: *Knowledge Management and Organizational Learning*. New York: Springer, 2009. p. 3–13.

LOURENÇO, G. C. U. et al. Os Artefatos De Software Enquanto Produtos Do Conhecimento: Uma Investigação Na Indústria De Software. VII Congresso Internacional de Conhecimento e Inovação. Anais...Guadalajara: EGC, 2018

MATTHIENSEN, A. *Uso do Coeficiente Alfa de Cronbach em Avaliações por Questionários*. [s.l.] Embrapa, 2011.

NAWINNA, D. P. A model of Knowledge Management: Delivering competitive advantage to small & medium scale software industry in Sri Lanka. 2011 6th International Conference on Industrial and Information Systems, ICIIS 2011, p. 414–419, 2011.

PATTON, M. Q. *Qualitative evaluation and research methods*. London: Sage Publications, 2015.

SANTOS, G. S. et al. Análise das atividades de gestão do conhecimento entre extensionistas e empresas incubadas: estudo de caso da incubadora da UNESC. *Revista de Extensão*, v. 1, p. 90-1-7, 2016.

SANTOS, T. G. DA S. et al. A inter-relação entre os elementos formadores do capital intelectual no Hospital Santa Casa de Maringá. *Revista Tecnologia e Sociedade*, v. 14, n. 33, p. 68-85, 2018.

SOMMER, R.; SOMMER, B. *A practical guide to behavioral research: tools and techniques*. Oxford: Oxford University Press, 2002.

SOUSA, B.; VITOR, R.; CESAR, P. IDENTIFICAÇÃO DE PERÍODOS HIDROLÓGICOS POR MEIO DE ESTATÍSTICA DESCRITIVA E DOS TESTES. *GEOSABERES: Revista de Estudos Geoeducacionais*, v. 6, n. 12, p. 113-129, 2015.

TENÓRIO, N. et al. Tool Based on Knowledge Management Process: an Interview Protocol To Gather Functional Requirements From Software Industry Experts. *MATTER: International Journal of Science and Technology*, v. 3, n. 1, p. 45-54, 1 jan. 2017.

TIM, B.-L.; HENDLER, J.; LASSILA, O. The Semantic Web. *Scientific Am*, v. May, p. 34-43, 2001.

VASANTHAPRIYAN, S. et al. An ontology-based knowledge sharing portal for software testing. 2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). *Anais...2017*

WETHERELL, M.; TAYLOR, S.; YATES, S. J. *Discourse Theory and Practice: a reader*. London: Thousand Oaks & New Delhi, 2001.

Recebido: 22/09/2020

Aprovado: 11/03/2022

DOI: 10.3895/rts.v18n51.13211

Como citar: OLIVEIRA, M.; TENÓRIO, N.; BORTOLOZZI, F. A compreensão de reporte de bugs no desenvolvimento e uso de software: uma representação do conhecimento por meio de ontologia. *Rev. Technol. Soc.*, Curitiba, v. 18, n. 51, p. 244-259, abr./jun., 2022. Disponível em: <https://periodicos.utfpr.edu.br/rts/article/view/13211>. Acesso em: XXX.

Correspondência:

Direito autoral: Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

