Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Sistemas de Informação

Apoiando a estimativa de esforço a partir de dados de reputação de desenvolvedores

Leojayme Rodrigues Manso Silva

JUIZ DE FORA
DEZEMBRO, 2016

Apoiando a estimativa de esforço a partir de dados de reputação de desenvolvedores

Leojayme Rodrigues Manso Silva

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciências da Computação
Bacharelado em Sistemas de Informação

Orientador: José Maria Nazar David

JUIZ DE FORA
DEZEMBRO, 2016

Apoiando a estimativa de esforço a partir de dados de reputação de desenvolvedores

Leojayme Rodrigues Manso Silva

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

José Maria Nazar David
Professor

Marco Antônio Pereira Araújo
Professor

Cláudio Augusto Silveira Lélis
Mestrando

JUIZ DE FORA

09 DE DEZEMBRO, 2016

Resumo

O presente trabalho visa tratar de como dados sobre a reputação de desenvolvedores podem impactar as estimativas realizadas no contexto de evolução e manutenção onde o desenvolvimento acontece de maneira distribuída e colaborativa. Para o desenvolvimento deste trabalho considera-se que os desenvolvedores têm acesso aos dados referentes as requisições de mudança e dados históricos do projeto. Entende-se por dados históricos de projetos todos os artefatos gerados durante a execução do mesmo juntamente com a documentação referente às decisões tomadas no decorrer da execução do mesmo.

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

DCC Departamento de Ciência da Computução

UFJF Universidade Federal de Juiz de Fora

1 Introdução

Devido ao grande avanço tecnológico, as empresas passaram a atuar de maneira global expandindo suas fronteiras e potencializando suas relações comerciais e profissionais, antes limitadas pela distância (NASCIMENTO; DAVID; CARNEIRO, 2013).

Este fato também influenciou na maneira como o software é produzido trazendo uma nova realidade para as empresas produtoras de software (NASCIMENTO; DAVID; CARNEIRO, 2013). A fim de atender às crescentes necessidades e manter o foco em requisitos de produtividade e qualidade, com baixo custo, as empresas têm distribuído geograficamente suas atividades (JOSANG; HALLER, 2007). Isto deve-se ao fato de que, com o passar do tempo, os softwares a serem desenvolvidos ficaram cada vez mais complexos, e tornaram-se um importante ativo para as organizações. Nesse contexto surge uma nova abordagem para desenvolvimento de software, o desenvolvimento de software distribuído. Nesta abordagem, as equipes encontram-se geograficamente distribuídas e trabalham de maneira colaborativa nas atividades de desenvolvimento de software (NASCIMENTO; DAVID; CARNEIRO, 2013). Esta prática traz benefícios que frequentemente estão ligados a custo, qualidade e produtividade, uma vez que, uma empresa pode querer montar uma equipe especialista procurando por profissionais de menor custo mantendo o mesmo nível de qualidade. O fato destes profissionais poderem trabalhar de maneira geograficamente dispersa facilita na construção dessa equipe.

O fato de existirem diversos colaboradores trabalhando de maneira geograficamente distribuída pode trazer riscos para as organizações, principalmente se existirem diversas organizações envolvidas (NAYAK; SUESAOWALUK, 2008). Nayak e Suesaowaluk (2008), identificaram como sendo os principais riscos envolvendo o desenvolvimento distribuído: a falta de confiança, divergências referentes a valores corporativos e um processo falho de compartilhamento de informação. Ainda levantou a importância de todos os profissionais possuírem uma visão comum do projeto em que estão trabalhando. Junto a isso levantou algumas diferenças entre o desenvolvimento tradicional e o distribuído sendo elas: cultura organizacional, gerenciamento, plataforma técnica e equipes de desen-

volvimento e problemas culturais e sociais. Essas diferenças dificultam o gerenciamento dos riscos devido à distribuição das das atividades envolvidas.

Para que a aplicação da abordagem de desenvolvimento distribuído seja bem sucedida os profissionais envolvidos devem confiar uns nos outros. O estabelecimento desta confiança pode ser dificultado quando os profissionais não se conhecem, fato que pode acontecer no contexto de equipes geograficamente distribuídas. A confiança pode ser entendida como a credibilidade que um indivíduo atribui a outro na realização de uma determinada ação (AL-ANI; REDMILES, 2009).

Nesse contexto, a reputação é um dos fatores que afetam a colaboração entre os membros das equipes. Reputação pode ser usada como indicador de cooperação. Logo, informações de reputação se tornaram um importante ativo para as empresas de desenvolvimento de software (LÉLIS et al., 2016). Para apoiar a construção da confiança entre indivíduos, sistemas de reputação (JOSANG; HALLER, 2007) (JØSANG, 2007) (JØSANG; QUATTROCIOCCHI, 2009) vêm sendo explorados como forma de fornecer informações que ajudem no estabelecimento de relações entre indivíduos desconhecidos. A reputação de um indivíduo pode ser considerada como a opinião geral de uma comunidade em relação ao comportamento do mesmo dentro de um determinado contexto(ZACHARIA; MOUKAS; MAES, 2000). A construção dessa reputação se dá pela interação entre indivíduos. As informações geradas são processadas e, no final, são disponibilizadas para servirem de suporte à decisão de outros indivíduos quanto a se relacionarem com outros indivíduos. Sistemas que realizam este propósito são ditos sistemas de reputação (JØSANG, 2007).

O desenvolvimento de software distribuído traz consigo alguns desafios (NASCIMENTO; DAVID; CARNEIRO, 2013). Parte desses desafios estão ligados a falta de mecanismos de comunicação mais flexíveis e informais que possuam recursos que apoiem de forma otimizada a comunicação síncrona e assíncrona (MOCKUS; HERBSLEB, 2001). A comunicação representa papel fundamental no processo de desenvolvimento de software (PRESSMAN, 2005). Em uma equipe na qual existem desenvolvedores distribuídos em diversos países, pode acontecer de, esses desenvolvedores não irão se conhecer pessoalmente. A comunicação informal face-a-face carrega em si informações contextuais que são mais difíceis de serem transportadas a meios formais(AL-ANI; REDMILES, 2009).

Através de uma revisão sistemática da literatura, Jalali e Wohlin (2010) revelam que nos últimos anos a abordagem de desenvolvimento de software distribuído aplicada em conjunto com metodologias ágeis cresceu de maneira significativa. As metodologias ágeis para desenvolvimento de software vêm sendo cada vez mais utilizadas por empresas devido ao aumento da complexidade do software a ser desenvolvido. Isto se dá devido a algumas características destas metodologias como: flexibilidade de escopo, uso de comunicação e processos informais e o foco nas pessoas em detrimento de processos (NASCIMENTO; DAVID; CARNEIRO, 2013). O surgimento destas metodologias também trouxe muitos desafios, sendo um dos principais a realização de estimativas de esforço para as atividades de desenvolvimento (MIGUEL et al., 2016). A realização de estimativas de esforço de maneira completamente arbitrária pode comprometer todo o planejamento de um projeto e refletir diretamente no prazo, na qualidade do software, bem como na satisfação do cliente. Embora abordagens tradicionais de estimativa de esforço sejam aplicadas a projetos que utilizam métodos ágeis, observa-se que as mesmas têm resultado em estimativas imprecisas (PAREDES; ANSLOW; MAURER, 2014). Informações importantes dos projetos podem ser perdidas quando não acontece o planejamento efetivo, e não se tem uma visão de desenvolvimento futura (PEIXOTO; AUDY; PRIKLADNICKI, 2010).

Tomando por base o contexto de empresas que distribuem suas atividades de desenvolvimento de software utilizando metodologias ágeis, o presente trabalho visa investigar como os dados de reputação de desenvolvedores de software podem auxiliar no processo de tomada de decisão, relacionada à uma estimativa de esforço em projetos de manutenção e evolução de software.

A seguinte hipótese foi formulada: *Se fornecermos um mecanismo baseado em visualizações, oferecendo indicadores sobre reputação dos desenvolvedores como forma de apoiar a tomada de decisão quanto à estimativas de esforço, então as decisões serão enriquecidas.*

A solução proposta consiste no desenvolvimento de um sistema baseado em visualizações que forneça informações sobre os dados de reputação de desenvolvedores. A partir dessas visualizações, colaboradores envolvidos no projeto podem discutir através de um fórum específico, e tomar decisões.

O presente trabalho está dividido em cinco capítulos. O segundo capítulo, apresenta o referencial teórico utilizado neste trabalho. Serão apresentados os principais conceitos e características sobre sistemas de reputação, desenvolvimento distribuído e visualização. No terceiro capítulo será apresentada a solução proposta pelo presente trabalho que consiste em realizar uma expansão da arquitetura ArchiRi (LÉLIS et al., 2016). No quarto capítulo será apresentada uma análise de uso da solução mostrando como a expansão proposta pode apoiar equipes de desenvolvimento no processo de tomada de decisão com base em dados de reputação de desenvolvedores. No último capítulo serão apresentadas as considerações finais.

2 Referencial teórico

Para o desenvolvimento do presente trabalho foi realizada uma pesquisa por trabalhos que abordem os temas: sistemas de reputação, desenvolvimento de software distribuído e visualizações de software. A busca por esses trabalhos foi realizada nas bases *IEEEXplore*, *ACM Digital Library* e *Scopus*.

2.1 Sistemas de reputação

Sistemas de reputação são softwares que permitem a coleta de informações para a construção da reputação. Esses softwares processam essas informações e as disponibilizam para um grupo de indivíduos interessados (NASCIMENTO; DAVID; CARNEIRO, 2013). A ideia básica de um sistema de reputação é permitir que diversos indivíduos atribuam um valor referente à reputação de outros indivíduos. Através da agregação destes valores atribui-se um valor final de reputação a cada um deles (JOSANG; HALLER, 2007). Uma consequência direta de se aplicar um sistema de reputação a ambiente onde a confiança é fundamental é a atenção de cada membro de uma comunidade onde o sistema está implantado em manter um bom comportamento para que alcance valores altos de reputação. Através dessa atitude, a qualidade das relações e serviços podem ser melhoradas em consequência do interesse em manter-se uma boa reputação (JOSANG; HALLER, 2007).

Um tipo de sistema de reputação é o binomial Bayesiano. Esse sistema se baseia em dois valores possíveis para a avaliação de uma pessoa, um positivo e um negativo. A grande desvantagem desse modelo é que desconsidera a possibilidade de avaliações em diversos níveis como um sistema de notas (JOSANG; HALLER, 2007). A partir de um estudo realizado sobre os sistemas de reputação binomial Bayesiano, os autores Josang e Haller propõe o modelo Dirichlet. O modelo Dirichlet baseia-se em avaliar um conjunto de elementos disjuntos para chegar ao valor final da reputação sendo assim um modelo multinomial. Essa distribuição foi escolhida para realizar o cálculo da reputação

pois pode-se assumir que não existe uma preferência de uma alternativa em relação a outra. Em geral, um sistema de reputação permite que agentes avaliem outros agentes e serviços em qualquer nível presente em um conjunto de níveis pré-definidos. Alguma condição de controle que permita uma pessoa realizar uma avaliação a respeito de outra deve ser definida para impedir que aconteçam avaliações sem uma interação prévia entre essas pessoas. Um exemplo de condição pode ser uma transação em um sistema de *e-commerce*. Agentes podem mudar seu comportamento no decorrer do tempo, logo faz sentido relacionar um peso maior a avaliações mais recentes. Para tal, inclui-se um fator referente à longevidade de uma avaliação.

2.2 Desenvolvimento distribuído de software

O software passou a desempenhar um papel importante dentro das organizações sendo papel fundamental para os processos de negócio (NASCIMENTO; DAVID; CARNEIRO, 2013). O software é considerado elemento básico dos sistemas de informação capturando e transformando dados em informação que, dada a sua relevância, torna-se elemento estratégico dentro das organizações (O'BRIEN; MARAKAS, 2010).

Dada a natureza intangível do software, diversas abordagens para o seu desenvolvimento foram experimentadas com intuito de produzi-lo eficientemente, dentro das expectativas relativas ao prazo, custo e qualidade (NASCIMENTO; DAVID; CARNEIRO, 2013). A engenharia de software vem apresentando como a aplicação de processos no desenvolvimento de software é importante quando se está em busca de fazê-lo de forma eficiente (PRESSMAN, 2005).

Para executar esse processo faz-se necessária a alocação de profissionais com habilidades específicas para desenvolver as atividades definidas. A partir disso, é formada uma equipe que, por exemplo, irá trabalhar nas regras de negócio do projeto e outra que irá trabalhar no desenvolvimento e testes. O fato dessas equipes existirem e trabalharem em conjunto caracteriza o desenvolvimento de software como uma atividade colaborativa (NASCIMENTO; DAVID; CARNEIRO, 2013).

Com a expansão das fronteiras das organizações fez-se possível a constituição de equipes com recursos que encontram-se geograficamente distribuídos. A adesão desta

prática de desenvolvimento é motivada pela possibilidade de expandir-se a área onde procura-se mão-de-obra especializada para a constituição de equipes, e ainda a possibilidade da redução de custos na produção do software.

Evaristo e Scudder (2000) propõem um modelo que aborda as dimensões que podem afetar a utilização dessa prática de forma que seja possível categorizar projetos que são desenvolvidos de maneira distribuída. O modelo proposto pelos autores é ilustrado pela figura 2.1.

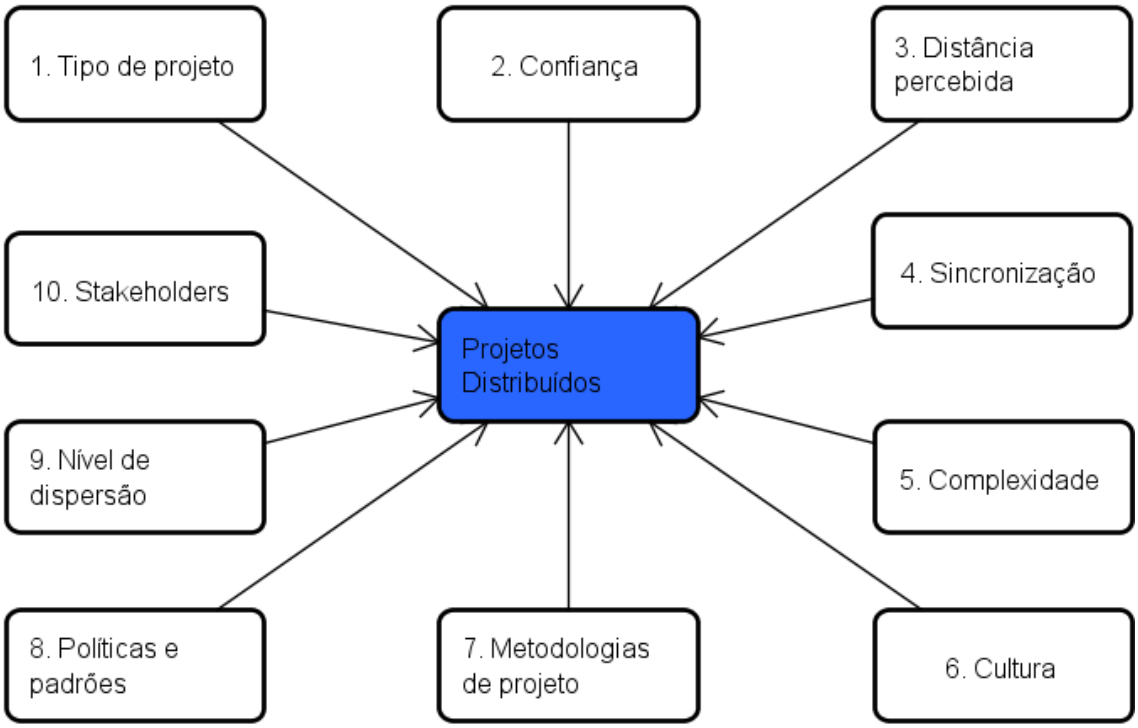

Figura 2.1: Dimensões do desenvolvimento distribuído (EVARISTO; SCUDDER, 2000)

1. Tipo de projeto

   O tipo de projeto afeta a forma como o mesmo deve ser gerenciado. Cada indivíduo que encontra-se distribuído pode ter uma abordagem diferente para realizar este gerenciamento.

2. Confiança

   A confiança é um fator de suma importância desta nova forma de trabalho uma vez que essa confiança influencia na coesão dos grupos e também no nível de colaboração entre as partes (LIPNACK; STAMPS, 1997). Quando existe um baixo nível de

confiança tende-se a despender um maior esforço em atividades de monitoramento.

3. Distância percebida

   Se refere à distância dos indivíduos envolvidos no projeto. Uma forma de realizar essa análise é através da possibilidade desses indivíduos realizarem a comunicação face-a-face. Dependendo dessa distância essa forma de comunicação pode ser inviável.

4. Sincronização

   A dimensão de sincronização está relacionada à possibilidade dos indivíduos que se encontram separadas geograficamente, poderem trabalhar em um mesmo projeto durante o mesmo período o que facilita o processo de comunicação.

5. Complexidade

   Segundo os autores que propuseram o modelo, a complexidade do software a ser desenvolvido influencia diretamente na produtividade das equipes uma vez que, quanto mais complexo seja o projeto, um número maior de profissionais distribuídos pode ser alocado aumentando assim o esforço para realizar a gestão dos mesmos.

6. Cultura

   Cultura por si só representa um fator multidimensional que afeta o desempenho de projetos distribuídos de diferentes maneiras. As variações desse fator podem afetar diferentemente o comportamento dos indivíduos envolvidos. Quando se têm indivíduos geograficamente distribuídos, este choque cultural precisa ser tratado de forma a alcançar uma forma de trabalho eficiente.

7. Metodologias de projeto

   Profissionais distribuídos podem estar acostumados a trabalharem seguindo diferentes metodologias, ou apenas possuem um nível de experiência diferente nas mesmas.

8. Políticas e padrões

   A dimensão de política e padrões está associada à capacidade dos profissionais em manter uma padronização em suas atividades bem como na produção dos artefatos

gerados no decorrer do projeto.

9. Nível de dispersão

   Esta dimensão é dada pela distância percebida entre um grupo de *stakeholders* que compõem um grupo maior formado por vários outros *stakeholders*. Outra medida a ser considerada é a distância entre os membros de um grupo específico, uma vez que, um grupo responsável por uma determinada atividade pode não estar disperso entre si, porém, estar disperso dos outros grupos.

10. *Stakeholders*

    Está associada ao grupo de *Stakeholders* que estão envolvidos no projeto onde cada um possui uma percepção do mesmo (PRIKLADNICKI et al., 2004). No contexto de desenvolvimento distribuído, são considerados como principais *Stakeholders*, os clientes, usuários e equipes. Tais *Stakeholders* podem estar no mesmo local ou distribuídos de maneira nacional, continental ou global (PRIKLADNICKI et al., 2003).

Nesse contexto, a comunicação representa um papel fundamental no processo de desenvolvimento de software (PRESSMAN, 2005). A comunicação informal face-a-face carrega em si informações contextuais que são mais difíceis de serem transportadas a meios formais (AL-ANI; REDMILES, 2009). A comunicação face-a-face é crucial para a sustentação das relações interpessoais, e a criação de elos é pré-condição para o compartilhamento de informações (HINDS, 2002).

Apesar das grandes dificuldades que esse tipo de desenvolvimento proporciona, o mesmo também traz consigo benefícios. Esses benefícios podem estar ligados a custo, qualidade e flexibilidade para o desenvolvimento (PRIKLADNICKI; AUDY; EVARISTO, 2003).

2.3 Visualização

Tavares et al. (2015) desenvolveu sua dissertação com o objetivo de desenvolver ou integrar soluções existentes com o intuito de apoiar a manutenção e a evolução de software no

contexto de desenvolvimento distribuído na tarefa de realizar manutenção em produtos de software.

Tavares et al. (2015) ressalta a importância da visualização no processo de compreensão de um conjunto de informações levando assim a um conhecimento mais profundo dessas informações. Foram utilizados dois paradigmas de visualização: AIMV (Ambiente Interativo de Múltiplas Visualizações), onde são implementadas múltiplas visões para um mesmo conjunto de dados e um que implementa diferentes visões para diferentes conjuntos de dados. No contexto de desenvolvimento realizado de maneira colaborativa, Isenberg et al. (2011) apresenta quatro definições para o conceito de visualização colaborativa, sendo elas:

- "Melhora a visualização tradicional, reunindo muitos especialistas, de modo que cada um pode contribuir para o objetivo comum de compreensão do objeto, fenômeno, ou dados sob investigação";
- "Refere-se a um subconjunto de aplicações CSCW em que o controle sobre parâmetros ou produtos do processo de visualização científica são compartilhados";
- "Permite que os usuários geograficamente separados acessem um ambiente compartilhado para visualizar e manipular conjuntos de dados para a resolução de problemas sem ter que se deslocarem fisicamente";
- Análise de dados sociais é o termo designado para classificar interações sociais que podem ocorrer numa visualização colaborativa. Alguns trabalhos tratam visualização colaborativa como análise de dados sociais, então a quarta definição apresentada fala a respeito disso: "Análise de dados social é uma versão de análise exploratória de dados que se baseia na interação social como fonte de inspiração e motivação".

Por considerar que essas definições não descreviam de forma abrangente o conceito de visualização colaborativa, Isenberg et al. (2011) elabora a seguinte definição: "Visualização colaborativa é o uso compartilhado e interativo de representações visuais de dados por mais de uma pessoa, apoiado por computador, com o objetivo comum de contribuir para as atividades de processamento de informação conjunta."

Novais et al. (2013) realizaram um mapeamento sistemático sobre a evolução da visualização de software.Nesse estudo eles puderam apurar que os paradigmas de visualização mais utilizados são as baseadas em gráficos, as estruturadas por hierarquia e projeções geométricas. Ainda observaram que a maioria dos trabalhos encontrados utilizam mais de uma técnica para trabalhar a visualização.

Em relação aos atributos visuais, o mais utilizado é a cor. Novais et al. (2013) atribuem esse fato a facilidade de implementação em ferramentas de software e a percepção humana às variações do atributo cor.

Já em relação à visualizações, Novais et al. (2013) ficaram surpresos com o resultado do mapeamento que mostrou que mais da metade dos estudos encontrados utilizam uma única visualização para apresentar dados através de gráficos. Porém não ficou comprovado que a abordagem utilizando uma única visualização produz melhores resultados do que a abordagem que utiliza múltiplas visualizações. Apesar de usarem apenas uma visualização, os estudos encontrados durante o mapeamento realizado por (NOVAIS et al., 2013) utilizam mais de um paradigma de visualização.

Novais et al. (2013) também investigaram os mecanismos de interação utilizados nos estudos encontrados durante o mapeamento. Esses mecanismos são utilizados para filtrar dinamicamente e selecionar os dados que estão sendo visualizados.

O desenvolvimento de ArchiRiXCom se baseou na definição de visualização colaborativa elaborada por Isenberg et al. (2011) utilizando o paradigma de visualização baseado em gráficos explorando o atributo visual cor.

2.4 Trabalhos relacionados

Durante o período de pesquisa pôde-se perceber que no contexto de desenvolvimento distribuído, as atividades de comunicação podem ser dificultadas dada a distância geográfica existente entre os indivíduos envolvidos. Outro fator considerado importante neste contexto, levantado por Lipnack e Stamps (1997), é a confiança, uma vez que, a mesma afeta a colaboração. Lélis et al. (2016) levanta a importância da reputação no contexto de desenvolvimento distribuído uma vez que a reputação também pode afetar o nível de colaboração entre os membros da equipe.

Ramarao et al. (2016) fazem uma análise de como a reputação de um relator de *bugs* pode impactar no tempo de correção dos *bugs* reportados. O número de *bugs* de um software tem um papel vital como quantificador de qualidade de um software, sendo um fator mais confiável quando comparado a outros fatores subjetivos fornecidos pela indústria, como por exemplo, a facilidade de uso (RAMARAO et al., 2016). O tempo acumulado para correção de um *bug* em uma atualização de um software é considerado um dos mais significativos indicadores de qualidade (KIM; JR, 2006). Ramarao et al. (2016) utilizaram dados extraídos da ferramenta Jira, que é utilizada como ferramenta para esta atividades no projeto JBoss. Para construir o modelo de análise os autores identificaram alguns parâmetros dos *bugs* reportados considerados cruciais e que impactam no tempo de correção sendo eles: Título, Descrição, Frequência, Severidade e Nome do relator.

Geralmente acredita-se que *bugs* similares irão precisar aproximadamente de uma mesma quantidade de tempo para serem resolvidos (RAMARAO et al., 2016). Ramarao et al. (2016) se basearam em um modelo proposto por Weiss et al. (2007) que avalia a similaridades dos *bugs* reportados para construírem seu próprio modelo. Em seu modelo os autores implementaram o algoritmo "k *Nearest Neighborhood*" para avaliar a similaridade dos *bugs* reportados a partir de dados históricos.

Para atribuir a reputação dos relatores Ramarao et al. (2016) consideram que esse valor é definido pelo número de *bugs* que foram relatados por um determinado relator e que eventualmente foram corrigidos. A base de dados utilizada para o trabalho era composta por dados de noventa e um relatores e de quinhentos e sessenta e sete *bugs* relatados.Para classificar um *bug* quanto ao tempo gasto na sua correção foram utilizadas quatro classes de categorização sendo elas: *"Very fast fixed bugs" , "Fast fixed bugs", "Slowly fixed bugs" e "Very slowly fixed bugs"*.

Após feita a análise, os autores chegaram a uma melhoria de acurácia de quase treze por cento na estimativa de tempo para a correção utilizando o modelo proposto, levando em consideração a reputação do relator em detrimento ao modelo que considera apenas a similaridade dos *bugs* relatados.

Ainda que Ramarao et al. (2016) tenham realizado uma análise sobre o impacto da reputação sobre o tempo de correção de um *bug*, não abordaram a estimativa de

tempo, bem como não utilizaram mecanismos de visualização para apoiar a análise das informações de reputação.

Bosu e Carver (2014), realizaram uma análise quanto o impacto da reputação dos desenvolvedores no resultado da revisão de código em projetos *OpenSource*. A construção de uma identidade e uma boa reputação foram levantados como os fatores motivadores que levam desenvolvedores a trabalhar em tais projetos.Para realizar a análise os autores separaram os desenvolvedores em dois grupos, sendo eles: desenvolvedores de núcleo e desenvolvedores periféricos.Os desenvolvedores do núcleo do projeto são aqueles que têm uma relação a mais tempo com o projeto e que já fizeram grandes contribuições para o projeto. Os autores assumem em seu trabalho que esses desenvolvedores possuem uma melhor reputação em relação aos periféricos.Os desenvolvedores periféricos são aqueles que contribuem eventualmente para o projeto e interagem com os desenvolvedores do núcleo e raramente com outro periférico.

Projetos *OpenSource* mais maduros consideram obrigatória a revisão de código, se for evidenciado que a reputação do desenvolvedor afeta essa tarefa pode-se considerar que afeta o projeto como todo.

Os pontos explorados na análise foram: Tempo para *feedback*, Intervalo de revisão, Aceitação do código e Número de *patches*.

Considera-se o tempo para *feedback* o intervalo de tempo entre a submissão do código para revisão até o primeiro comentário de revisão. Alguns fatores podem sugerir que os desenvolvedores de núcleo possuam esse tempo de *feedback* menor, tais como: conhecer o membro da equipe mais capacitado para realizar a revisão e podem o adicionar como revisor potencial; a posição deles pode levar revisores a consumirem menos tempo de revisão; a prioridade de suas interações e bom relacionamento pode levar outros membros a priorizarem suas solicitações.

O intervalo de revisão é o intervalo entre o início e o fim do processo de revisão. Foi definido neste trabalho que o processo termina quando um *patch* é adicionado ao projeto principal. A aceitação do código é obtida através do número de solicitações de revisão de código e o número de *patches* adicionado ao projeto principal. O número de *patches* está ligado ao número de patches que foram necessários que um desenvolvedor

gerasse para que sua revisão fosse concluída.

Para realizar o seu trabalho os autores utilizaram dados de diversos projetos *OpenSource* dentre eles: *Chromium OS, LibreOffice e OpenStack*.

Bosu e Carver (2014) avaliaram como a reputação pode influenciar na atividades específica de revisão de código dentro de um projeto *OpenSource*. Embora esse tipo de projeto seja desenvolvido de maneira distribuída e colaborativa, os autores não exploram elementos de visualização e não oferece suporte a tomada de decisões.

No contexto de desenvolvimento distribuído de software Nascimento e Santoro (2009) apresentam um modelo no qual desenvolvedores se relacionam e produzem código fonte. Esse modelo é centrado nos conceitos de interação e nos artefatos produzidos. Um participante é identificado pelo seu perfil e sua reputação. Essa reputação é apresentada explicitamente para os participantes que também tem o apoio do meio de comunicação que armazena as mensagens trocadas entre os desenvolvedores. Assim, o modelo relaciona um artefato a um participante pela sua avaliação e identifica a interação ocorrida para gerar o artefato.

O trabalho desenvolvido por Nascimento e Santoro (2009) trata da reputação dos desenvolvedores com o foco na colaboração, porém, não avança na utilização dos elementos de visualização como forma de apoio à tomada de decisões.

Sänger e Pernul (2014) desenvolveram em seu trabalho uma proposta do uso de visualização em sistemas de reputação. Através dessa proposta os autores visam permitir uma exploração visual da reputação por meio de uma apresentação gráfica de referenciais e seus atributos de contexto. Os referenciais seriam as fontes de informação e os atributos seriam os critérios e os valores atribuídos a cada um desses critérios. Dessa forma espera-se alcançar a transparência no processo de atribuição e cálculo de reputação já que o usuário pode verificar os dados de cada fonte.

Apesar de explorar mecanismos de visualização para informações de reputação, o trabalho desenvolvido por Sänger e Pernul (2014) não explora os fatores comunicação entre os usuários nem a tomada de decisões, fatores que são explorados na ArchiRiXCom.

O trabalho desenvolvido por Kim et al. (2015) foi desenvolvido no contexto de transações *online*.Kim et al. (2015) defendem que uma terceira força pode guiar os can-

didatos a efetivar uma transação comercial provocando discussões e provendo opiniões sobre cada um. Foram utilizadas visualizações gráficas 3D para apresentar a reputação através de métricas WOM (*Word of Mouth*) geradas a partir do *feedback* dos usuários. Isso é utilizado para reorganizar o grupo e aprimorar o nível de conhecimento fornecido pelos especialistas desse grupo.

Kim et al. (2015) utilizaram mecanismos de visualizações para representar informações de reputação. Porém, esse trabalho foi desenvolvido no contexto de transações *online* e não permitem o intercâmbio das informações para o contexto de desenvolvimento distribuído de software.

3 ArchiRiXCom

Baseado nesse contexto, a solução propõe o desenvolvimento de um sistema baseado em visualizações que apoie a tomada de decisões dos membros de equipes que trabalhem em projetos de desenvolvimento de software de maneira colaborativa e distribuída.

O objetivo é fornecer visualizações que apoiem membros de equipes no processo de tomadas de decisão referentes a estimativas de esforço. Um objetivo secundário é fornecer um mecanismo de comunicação onde esses membros de equipes possam discutir e enriquecer as decisões tomadas.

Para o desenvolvimento desta solução foi feito um estudo sobre a arquitetura ArchiRI (LÉLIS et al., 2016) que apresenta um ferramenta onde é possível realizar uma análise quanto a reputação dos indivíduos de maneira individual, dentro de um grupo e em um determinado contexto apoiados por uma ontologia.

Após realizada essa análise percebeu-se que a mesma oferece visualizações que permitem a realização de uma análise sobre o contexto em que os indivíduos estão inseridos, a qual grupo que os indivíduos pertencem, sobre os valores para cada parâmetro que define a reputação de um indivíduo e a comparação desses parâmetros para cada indivíduo. Porém, a ArchiRi (LÉLIS et al., 2016) não possui nenhum módulo voltado para a comunicação entre os indivíduos bem como não possui visualizações que permitam uma análise histórica da reputação desses indivíduos nem uma visualização de *ranking* de reputação. A partir dessa análise percebeu-se uma oportunidade de expansão da arquitetura ArchiRi (LÉLIS et al., 2016) trazendo novas visualizações e implementando um módulo de comunicação que permita a interação entre os indivíduos.

Na seção 3.1 será apresentada a arquitetura ArchiRi(LÉLIS et al., 2016), na seção 3.2 será apresentada a arquitetura da ArchiRiXCom definida para a realização da expansão da ArchiRi na seção 3.3 serão apresentados os requisitos funcionais levantados para realizar a expansão de ArchiRI no formato de diagrama de casos de uso juntamente com a descrição de cada um deles, na seção 3.4 serão apresentados os requisitos não funcionais levantados para realizar tal expansão, na seção 3.5 será apresentado o modelo

de entidades e relacionamentos do banco de dados com as tabelas que foram necessárias para realizar essa expansão, na seção 3.6 será apresentado o modelo usado para o cálculo da reputação e por fim na seção 3.7 serão apresentados alguns detalhes de implementação juntamente com o ambiente de desenvolvimento utilizado para implementação da solução bem como para a construção dos modelos.

3.1 ArchiRi

A arquitetura ArchiRI foi criada com o intuito de gerenciar informações de reputação interoperáveis no contexto do projeto GiveMe Infra (TAVARES et al., 2015). Para isso propõem uma solução baseada na interoperabilidade e na importação e exportação de dados apoiada por uma ontologia, além de facilitar a visualização desses dados e a tomada de decisões a partir dos mesmos. A partir da ontologia desenvolvida por Casare e Sichman (2005), uma nova ontologia foi estabelecida, a ArchiRiOnt a partir de um modelo de reputação de três níveis, sendo eles: entidade, grupo e domínio.

Na primeira camada o interesse é apresentar dados de um entidade. Uma entidade pode ser um indivíduo, um serviço, um agente ou um produto. Considera-se que um indivíduo possui características próprias e avalia outros indivíduos bem como é avaliado pelos mesmos.

Estes indivíduos podem se relacionar, trocar informações e ter opiniões semelhantes, podendo assim ser agrupadas e dar origem a grupos. Na segunda camada considera-se que diferentes grupos possuem características específicas que os distinguem. Ainda, um grupo pode avaliar outros grupos bem como ser avaliados dando origem assim a reputação do grupo como um todo. Neste nível é possível analisar com quem um determinado indivíduo formou um grupo através da relação “Source” e “Target” onde o primeiro será responsável por oferecer sua opinião sobre o segundo. Para estabelecer essa relação, “Source” e “Target” devem pertencer ao mesmo domínio.

A camada de domínio é referente ao domínio ou contexto em que grupos e indivíduos estão inseridos. Em diferentes contextos grupos e indivíduos podem assumir diferentes reputações.

Nesta arquitetura considera-se que o valor de reputação em uma determinada

camada pode ser considerado como critério para definição da reputação nas outras camadas.A arquitetura foi projetada sendo composta por componentes internos e externos. A figura 3.1 apresenta esta arquitetura e seus principais módulos.

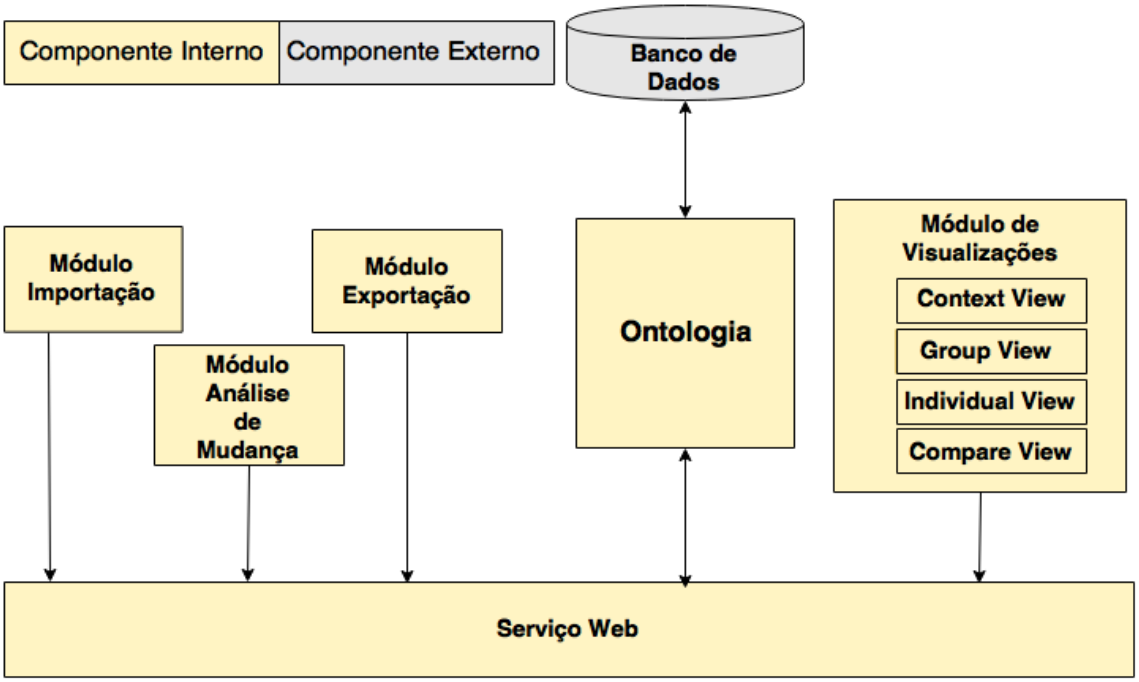

Figura 3.1: Arquitetura ArchiRi (LÉLIS et al., 2016)

Abaixo segue o detalhamento de alguns dos principais módulos desta arquitetura.

1. Serviço Web

   É responsável pelo contexto da ontologia. É capaz de consumir os dados inferidos pela ontologia e atualizá-los. Em contrapartida a ontologia utiliza métodos de manipulação de banco de dados para inserir e alterar informações. Na execução de ArchiRI, banco de dados e ontologia formam um servidor contendo informações de reputação. Quando um usuário solicita informações às inferências são acionadas. A partir do momento em que o usuário refinar sua pesquisa novos dados podem ser coletados e o sistema pode executar novas inferências.

   O banco de dados está dividido entre dados internos e externos. Os dados internos são aqueles gerados pela própria ArchiRI. Já os externos são originados, mantidos e gerenciados pela GiveMe Infra (TAVARES et al., 2015).

2. Importação e exportação

Através deste módulo ArchiRI promove o compartilhamento de informações, a portabilidade e a interoperabilidade. A importação pode ser usada para alimentar a base do sistema ou analisar dados de uma entidade em específico. Ao importar dados para o sistema a ontologia é acionada e inferências são executadas sobre os dados recém adicionados. A partir daí é possível identificar grupos e contexto aos quais a entidade pertence. O sistema é capaz de exportar as informações mostradas pela visualização Individual View nos formatos JSON, XML e TXT.

3. Análise de troca

   Este módulo oferece duas maneiras possíveis para a troca de informações de reputação. A primeira tem foco no contexto ao qual a entidade pertence e o segundo possui foco no grupo. Na primeira o objetivo é identificar os contextos aos quais as entidades poderiam pertencer juntamente com os critérios que justificam essa mudança. Já no segundo caso, o objetivo é identificar grupos com contextos semelhantes mostrando quais grupos poderiam receber a entidade analisada e quais critérios seriam utilizados para o alinhamento dos grupos.

4. Visualização

   Este módulo utiliza a biblioteca JavaScript d3js[1] para representar diferentes tipos de informação com diferentes possibilidades de visualização, facilitando a interpretação dos dados pelo usuário.

   Este módulo é composto por quatro visualizações, sendo elas:

   - Context View

     Através desta visualização é possível identificar todos os membros que pertencem ao mesmo contexto de um determinado membro especificado através do sistema.

   - Group View

     A visualização de grupo permite identificar todos os membros que pertencem ao mesmo grupo que outro membro especificado através do sistema.

[1] https://d3js.org/

- Individual View

  Esta visualização tem como o objetivo de apresentar os critérios e os valores que determinaram a reputação final da entidade.

- Compare View

  Foi criada com a finalidade de apresentar uma comparação dos critérios e seus valores, que determinam a reputação de várias entidades.

3.2 Arquitetura ArchiRiXCom

Assim como a arquitetura de ArchiRi (LÉLIS et al., 2016) foi projetada sendo composta por componentes internos e externos, a ArchiRiXCom seguiu o mesmo projeto. A figura 3.2 apresenta o resultado da expansão realizada na arquitetura e seus principais módulos os diferenciando entre os que já existiam na ArchiRi (LÉLIS et al., 2016) dos adicionados pela ArchiRiXCom.

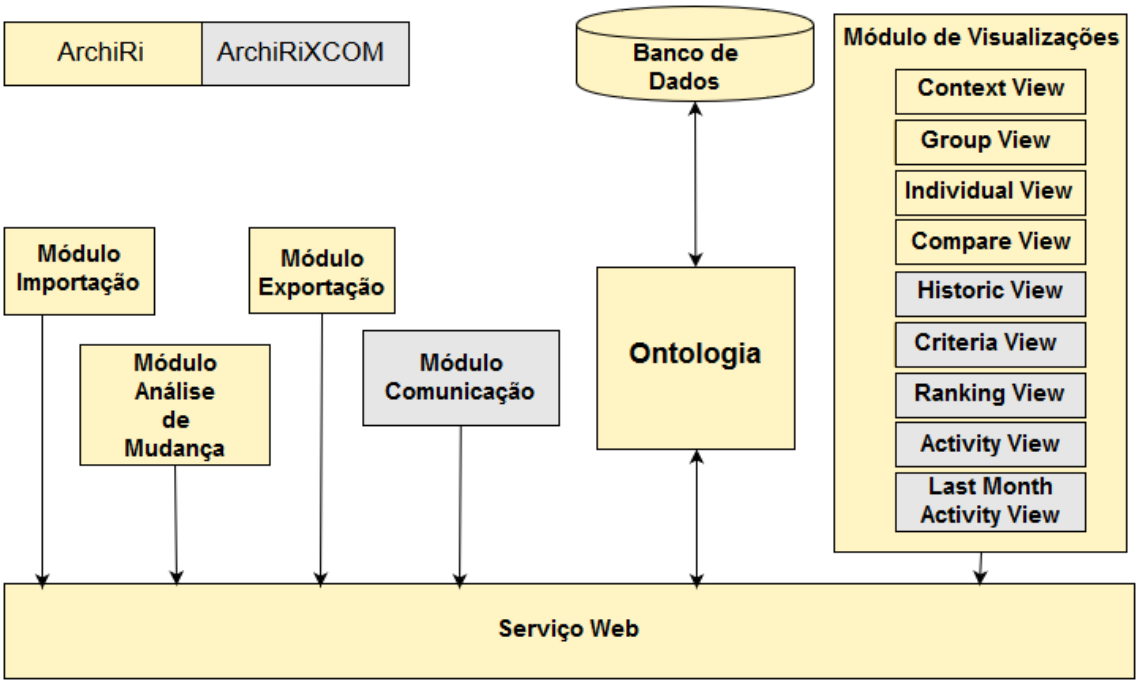

Figura 3.2: Arquitetura ArchiRiXCom

3.3 Requisitos Funcionais

A partir de uma análise realizada sobre a arquitetura ArchiRi (LÉLIS et al., 2016) e sobre o contexto apresentado na seção de introdução, foi realizada uma etapa de definição de requisitos visando expandir a arquitetura fornecendo novas visualizações sobre informações de dados de reputação que irão servir de apoio à tomada de decisões.

A figura 3.3 ilustra o diagrama de casos de uso o qual baseou o desenvolvimento da solução proposta. Nesse diagrama o ator usuário representa as entidades desenvolvedores, gerentes e quaisquer outros integrantes de uma equipe que possam ser parte interessada nas informações oferecidas pela ferramenta.

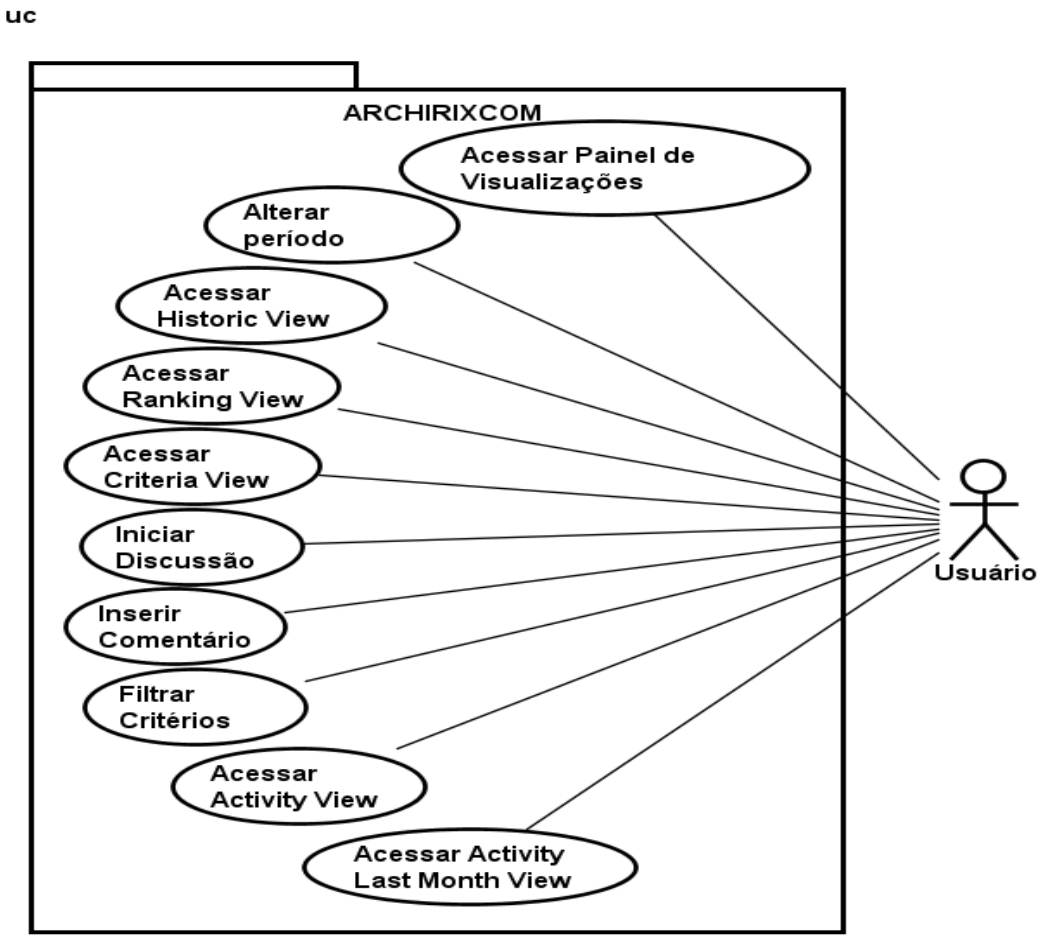

Figura 3.3: Diagrama de casos de uso ArchiRiXCom

3.3.1 Casos de uso do Usuário

Abaixo segue a descrição dos casos de uso presentes no diagrama representado pela figura 4.2.

1. Acessar Painel de Visualizações

   Um usuário terá acesso a um painel de visualizações no estilo *dashboard* onde poderá visualizar gráficos comparando informações de reputação dos desenvolvedores, analisar como a reputação de cada entidade evoluiu de acordo com o tempo e os critérios que levaram a reputação de cada entidade até o momento.

2. Acessar Historic View

   Um usuário terá acesso a uma visualização onde poderá analisar com a reputação de cada entidade evolui com o passar do tempo bem como ainda nesta visualização terá acesso aos tópicos de discussão criados pelas entidades e os comentários gerados para cada tópico.

3. Inicia Discussão

   Um usuário será capaz de criar um tópico de discussão para qualquer uma das visualizações pertencentes ao sistema. Para tal, ele deve se identificar e escrever o comentário que deseja realizar. Após criar um tópico de discussão, todos os usuários que acessarem a visualização em questão poderão visualizar e comentar o mesmo.

4. Inserir Comentário

   Um usuário será capaz de escrever um comentário para tópicos de discussão previamente criados por ele ou por outras entidades do sistema. Para tal, o usuário deve selecionar a opção de comentar existente em um tópico de discussão, identificar-se e escrever o seu comentário. Após concluir o seu comentário o mesmo ficará visível para todos os outros usuários do sistema.

5. Alterar período

   Um usuário poderá alterar o período de tempo que deseja avaliar a evolução da reputação das entidades através de um filtro presente na própria visualização. Ele

poderá realizar tal ajuste tanto no painel de visualizações quanto ao acessar a visualização histórica.

6. Acessar Ranking View

   Um usuário será capaz de acessar uma visualização que irá exibir um comparativo entre as reputações de todas as entidades bem como terá acesso aos tópicos de discussão criados pelas entidades e os comentários gerados para cada tópico.

7. Acessar Criteria View

   Um usuário será capaz de acessar uma visualização que irá exibir os valores dos critérios para cálculo da reputação até o momento. Nesta visualização ele poderá ver estes valores para cada uma das entidades bem como terá acesso aos tópicos de discussão criados pelas entidades e os comentários gerados para cada tópico.

8. Filtrar Critérios

   Ao acessar a visualização de critérios, um usuário será capaz de filtrar quais critérios ele deseja que seja exibido na visualização. Para tal o usuário deve utilizar o filtro presente na própria visualização.

9. Acessar Activity View

   Um usuário será capaz de acessar uma visualização onde poderá analisar a atividade de cada uma das entidades do sistema no fórum de discussões.

10. Acessar Activity Last Month View

    Um usuário será capaz de acessar uma visualização onde poderá analisar a atividade de cada uma das entidades do sistema no fórum de discussões no decorrer dos últimos 30 dias.

3.4 Requisitos não funcionais

1. Confiabilidade

   O sistema deve ser capaz de tratar e se recuperar de eventuais falhas garantindo a integridade das informações de forma a apresentar dados confiáveis.

2. Usabilidade

   As interfaces do sistema devem ser projetadas de forma a permitir uma interpretação clara e subjetiva das informações fornecidas pelas visualizações.

3. Interoperabilidade

   As visualizações do sistema devem ser geradas através de Web Services permitindo a reutilização dessas visualizações por diferentes modelos de cálculo de reputação trazendo uma maior flexibilidade para a ferramenta.

3.5 Modelo de banco de dados

Após a fase de definição dos requisitos que a ArchiRiXCom deve atender fez-se necessária a etapa de modelagem do banco de dados com o objetivo de manter todos os dados necessários para gerar as visualizações propostas. As tabelas apresentadas pelo modelo definido foram adicionadas ao banco de dados interno da arquitetura ArchiRI (LÉLIS et al., 2016) ou uma que precisou ser alterada. As tabelas foram geradas e algumas alteradas de forma a não comprometer o funcionamento da arquitetura.

A figura 3.4 ilustra o modelo de entidades e relacionamentos do banco de dados para a solução proposta.

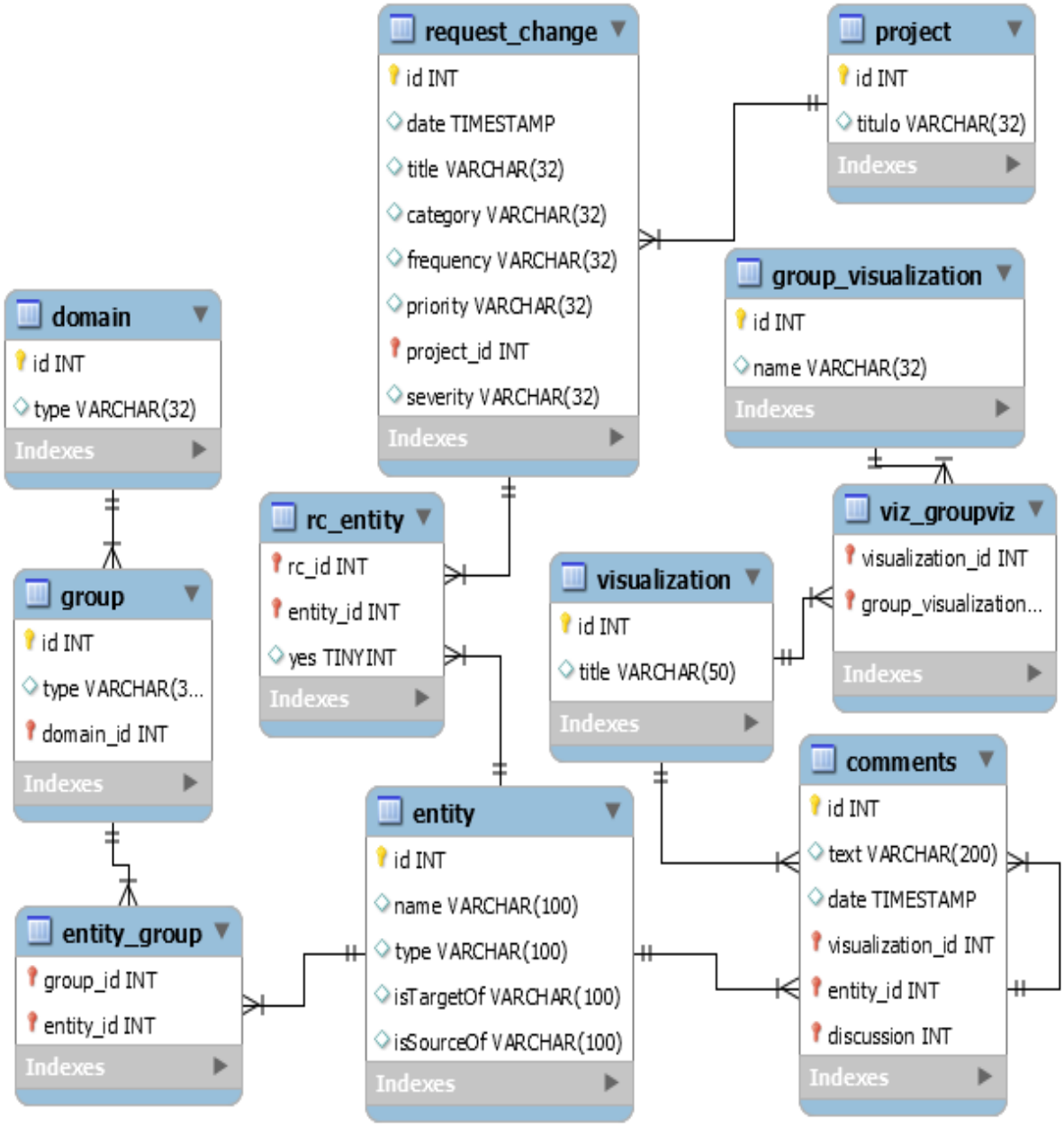

Figura 3.4: Modelo de entidades e relacionamentos ArchiRiXCom

Abaixo segue a descrição das tabelas bem como de seus relacionamentos e atributos.

- Domain

  Seguindo o modelo proposto por Lélis et al. (2016) de analisar a reputação das entidades em três níveis, a tabela 3.1 refere-se ao domínio ou contexto em que as entidades e grupos estão inseridas.

Tabela 3.1: Domain

Coluna	Tipo	Tamanho	PK	FK
id	inteiro		sim	não
type	varchar	32	não	não

- Group

  Ainda baseado no modelo para análise de reputação em três níveis proposto por Lélis et al. (2016), a tabela 3.2 representa os grupos ou equipes que podem existir. Este grupo é composto por um conjunto de entidades e se relaciona com a tabela domain, onde um grupo está inserido em apenas um domínio e um domínio pode possuir vários grupos inseridos nele.

Tabela 3.2: Group

Coluna	Tipo	Tamanho	PK	FK
id	inteiro		sim	não
type	varchar	32	não	não
domainid	inteiro		não	sim

- Entitygroup

  A tabela 3.3 foi criada para estabelecer o relacionamento muitos para muitos entre a tabela group e a tabela entity.

Tabela 3.3: Entitygroup

Coluna	Tipo	Tamanho	PK	FK
groupid	inteiro		não	sim
entityid	inteiro		não	sim

- Entity

  A tabela 3.4 foi criada para armazenar os dados referentes às entidades do sistema. Nela serão armazenados informações de identificação das entidades que podem ser

desenvolvedores, coordenadores, gerentes ou qualquer outro membro de uma equipe de desenvolvimento de software.

Tabela 3.4: Entity

Coluna	Tipo	Tamanho	PK	FK
id	inteiro		sim	não
name	varchar	100	não	não
type	varchar	32	não	não
isTargetOf	varchar	100	não	não
isSourceOf	varchar	100	não	não

- Rcentity

  Esta tabela 3.5 foi criada para estabelecer o relacionamento muitos para muitos entre as tabelas entity e requestchange.

Tabela 3.5: Rcentity

Coluna	Tipo	Tamanho	PK	FK
rcid	inteiro		não	sim
entityid	inteiro		não	sim
yes	booleano		não	não

- Requestchange

  Na tabela 3.6 são registradas as requisições de mudanças realizadas para um dado projeto.

Tabela 3.6: Requestchange

Coluna	Tipo	Tamanho	PK	FK
id	inteiro		sim	não
date	timestamp		não	não
title	varchar	32	não	não
category	varchar	32	não	não
frequency	varchar	32	não	não
priority	varchar	32	não	não
severity	varchar	32	não	não
projectid	inteiro		não	sim

- Project

  A tabela 3.7 project armazena informações sobre projetos que podem estar relacionados a diversas requisições de mudança.

Tabela 3.7: Project

Coluna	Tipo	Tamanho	PK	FK
id	inteiro		sim	não
titulo	varchar	32	não	não

- Visualization

  A tabela 3.8 armazena informações sobre as visualizações geradas pela solução proposta.

Tabela 3.8: Visualization

Coluna	Tipo	Tamanho	PK	FK
id	inteiro		sim	não
titulo	varchar	50	não	não

- Comments

A tabela 3.9 armazena informações sobre os comentários e as discussões criadas pelas entidades que utilizam o sistema. Esta tabela relaciona-se com a tabela entity e a tabela visualization, uma vez que, um comentário ou uma discussão está relacionada a uma entidade e a uma visualização.

Tabela 3.9: Comments

Coluna	Tipo	Tamanho	PK	FK
id	inteiro		sim	não
text	varchar	200	não	não
date	timestamp		não	não
visualizationid	inteiro		não	sim
entityid	inteiro		não	sim
discussion	inteiro		não	sim

3.6 Modelo de cálculo de reputação

Baseado no modelo Baeysiano apresentado por Jøsang e Quattrociocchi (2009) que têm por objetivo fornecer informação sobre reputação através do cálculo de um valor esperado para qualificações que um indivíduo pode receber em um contexto específico.

No contexto desta pesquisa, a probabilidade de um desenvolvedor ser atribuído a um requisição de mudança considerando os dados históricos de requisição de mudança. Existem duas situações possíveis quando uma equipe recebe uma nova requisição de mudança: é possível que um indivíduo seja atribuído a uma requisição de mudança recebendo um "sim" ou então este desenvolvedor não seja atribuído a uma requisição de mudança e receba um "não". O número total de requisições de mudança em que um desenvolvedor recebeu "sim " é representado pelo valor b(y), e representa o número de requisições de mudança que foram atribuídas a este desenvolvedor. O somatório de todas as requisições de mudança (qualificações "sim" mais "não") $\sum bdev(i)$ indica o número total de requisições de mudança recebidas pela equipe.

Esse modelo leva em consideração o fator tempo em relação às qualificações recebidas por um desenvolvedor permitindo uma análise quanto a mudança de comportamento

do mesmo no decorrer do tempo.

A constante W, chamada de a priori, representa o peso dado às qualificações anteriores no cálculo da reputação. Normalmente o valor de W é dado através da relação W = k onde k é o número de possíveis valores que uma qualificação pode assumir. Em nosso contexto o k = 2 ("sim" ou "não").

Ainda considera-se o valor de probabilidade inicial de uma qualificação ocorrer representada pela variável P. Considerando k = 2 e uma distribuição uniforme de probabilidade chegamos em P = 1/2. O cálculo da reputação é dado através da equação que segue abaixo.

$$Rdev = \frac{bdev(y) + W * P}{W + \sum bdev(i)} \tag{3.1}$$

Com valor de reputação calculado é possível estabelecer um *ranking* com os melhores desenvolvedores para cada requisição de mudança. A partir disto pode-se selecionar para o processo de estimativa somente aqueles que possuem uma valor mais alto de reputação. Acredita-se que desenvolvedores com uma reputação maior possuem uma confiança maior e uma melhor taxa de acerto na estimativa.

3.7 Implementação

A solução foi implementada de forma que todas as visualizações geradas possam ser utilizadas por outros modelos para cálculo de reputação, ou seja, ao alterar-se o modelo será as visualizações serão geradas exibindo novos valores quanto a reputação das entidades envolvidas.

As visualizações geradas pela ArchiRiXCom podem ser exportadas nos formatos: JPEG, PNG, SVG e PDF. Essa funcionalidade possibilita utilização das visualizações em trabalhos acadêmicos ou em apresentações onde as informações sobre reputação precisam ser exibidas mas não existe a possibilidade de apresentá-las pela ferramenta.

A figura 3.5 apresenta como a decisão tomada por um membro de uma equipe relaciona-se com as discussões referentes a um grupo de visualizações. Ainda mostra como o modelo responsável por gerar os dados de reputação destes membros relaciona-se com as visualizações, deixando claro que ao mudar-se o modelo serão geradas novas visualizações

para apresentar tais dados. Através desta interação dos membros de equipe baseada pelas visualizações a ArchiRiXCom apoia a tomada de decisões.

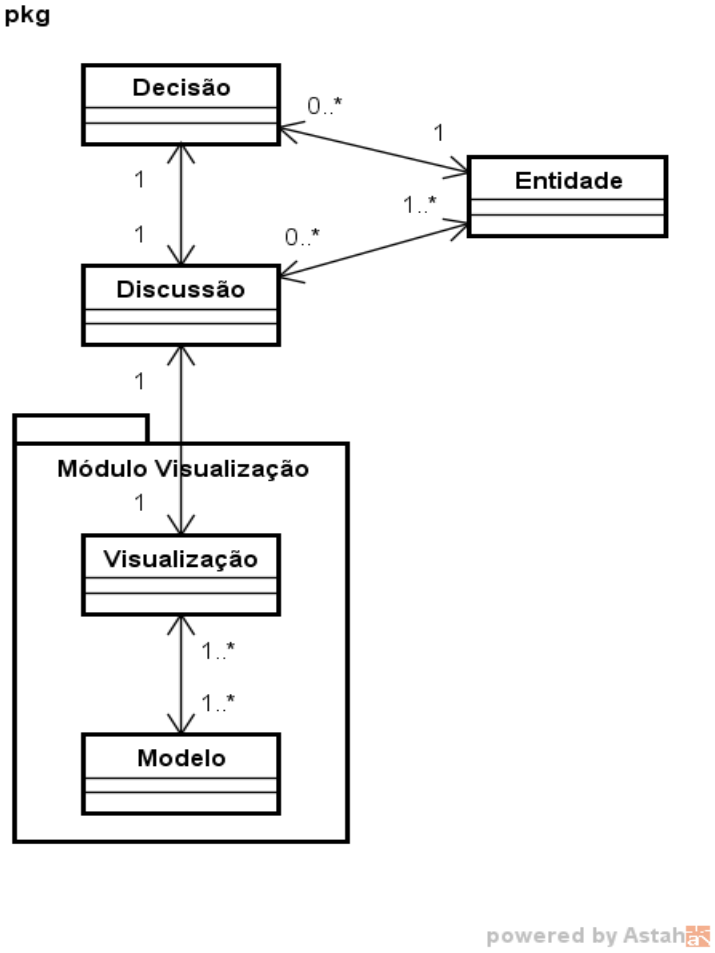

Figura 3.5: Diagrama conceitual ArchiRiXCom

Abaixo seguem as descrições de cada uma das visualizações que foram integradas à arquitetura ArchiRi (LÉLIS et al., 2016) ao realizar-se a expansão para ArchiRiXCom.

3.7.1 Historic View

A figura 3.6 apresenta os valores de reputação atribuídos a cada uma das entidades do sistema em um determinado momento. Através desta visualização pode-se observar a evolução da reputação de uma entidade permitindo uma análise do seu comportamento no decorrer do tempo.

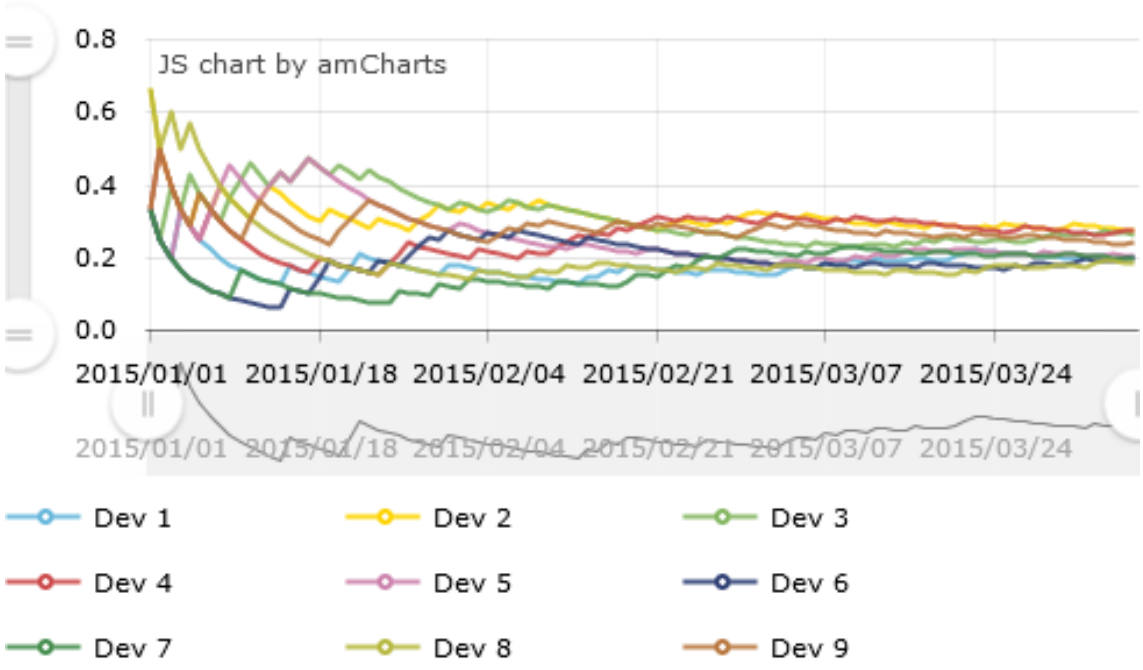

Figura 3.6: Historic View

O sistema dá a possibilidade para o usuário de filtrar os dados apresentados na visualização de acordo com um determinado período de tempo (eixo x do gráfico), ou então de filtrar de acordo com um intervalo de valor de reputação (eixo y). O sistema ainda permite que o usuário filtre os dados de quais entidades ele deseja visualizar através da legenda presente abaixo do eixo x.

3.7.2 Criteria View

Esta 3.7 apresenta o número de requisições de mudança atribuídas a cada entidade do sistema. Através dessa visualização pode-se analisar qual entidade recebeu mais atribuições de requisição de mudança. Isso pode impactar diretamente na reputação da mesma de acordo com o modelo proposto pelo presente trabalho.

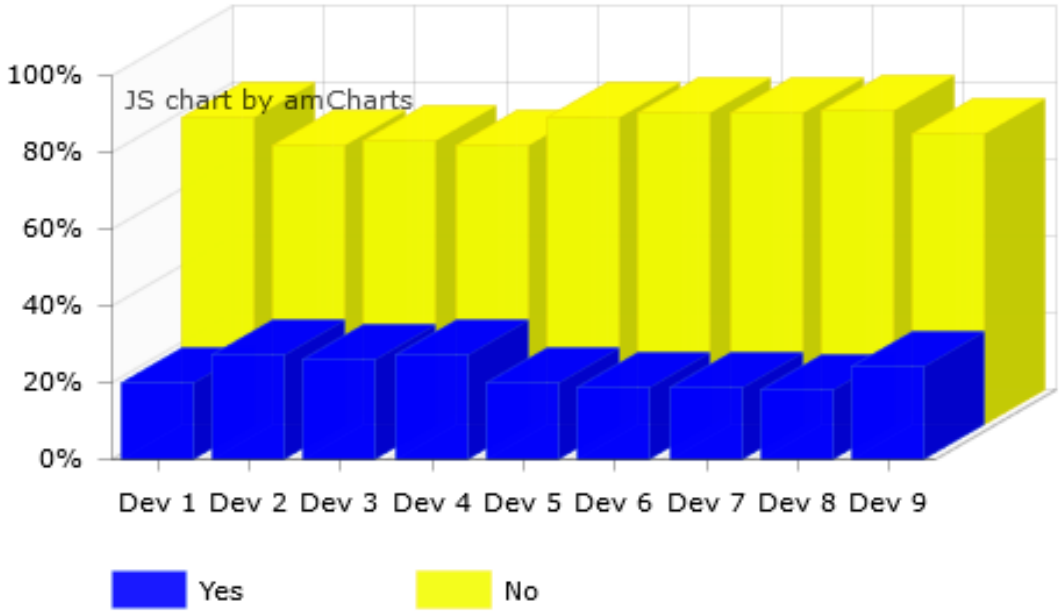

Figura 3.7: Criteria View

O sistema dá a possibilidade de o usuário filtrar os dados exibidos pela visualização. O usuário pode optar por visualizar apenas valores "yes" referentes a atribuições de mudança ou então valores "no".

3.7.3 Ranking View

A figura 3.8 apresenta um *ranking* referente a reputação de cada uma das entidades do sistema. Através dessa visualização pode-se analisar quais entidades possuem uma reputação maior no momento.

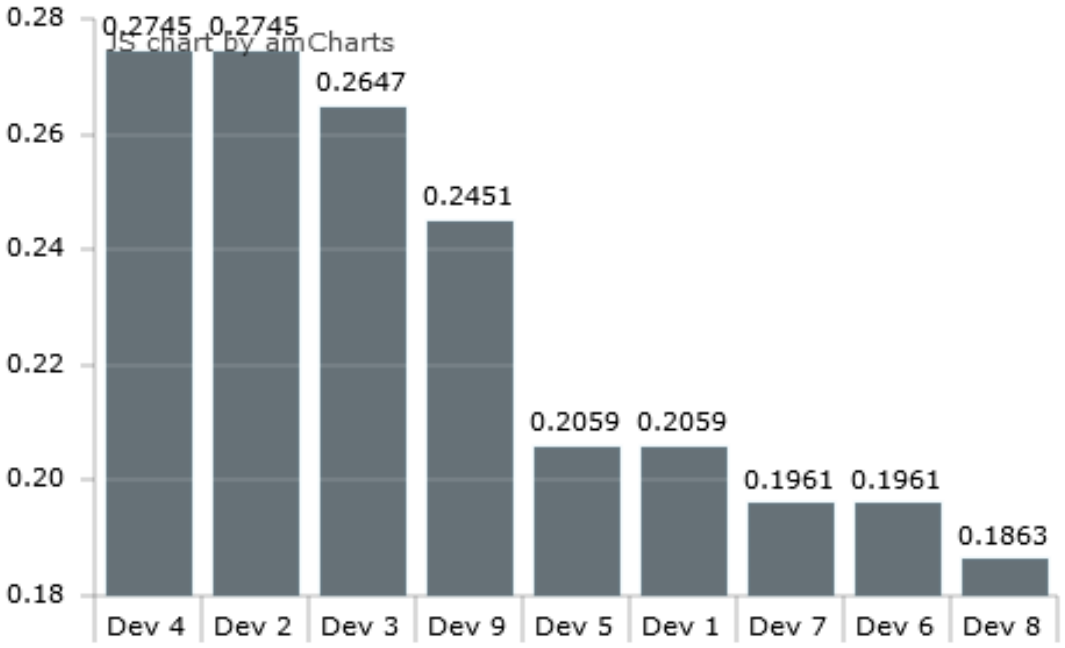

Figura 3.8: Ranking View

3.7.4 Activity View

Através da figura 3.9 pode-se analisar a participação de cada entidade do sistema nos fóruns de discussão. Esta visualização apresenta o número de discussões iniciadas por cada uma das entidades desde sua inserção no sistema, bem como, o número de comentários que cada uma gerou em discussões.

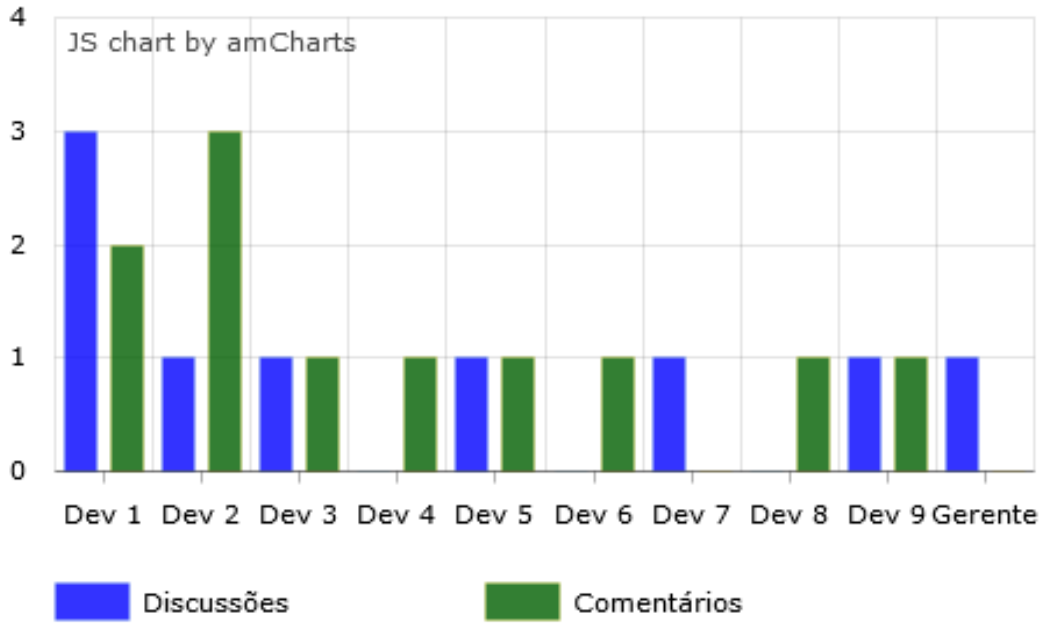

Figura 3.9: Activity View

3.7.5 Activity Last Month View

A figura 3.10 apresenta uma visualização que assemelha-se a visualização *Activity View*. Nessa visualização pode-se analisar participação de cada entidade do sistema nos fóruns de discussão nos últimos trinta dias. São apresentados dados quanto ao número de discussões iniciadas por cada entidade do sistema nos últimos 30 dias, bem como, o número de comentários realizados por cada entidade neste mesmo período.

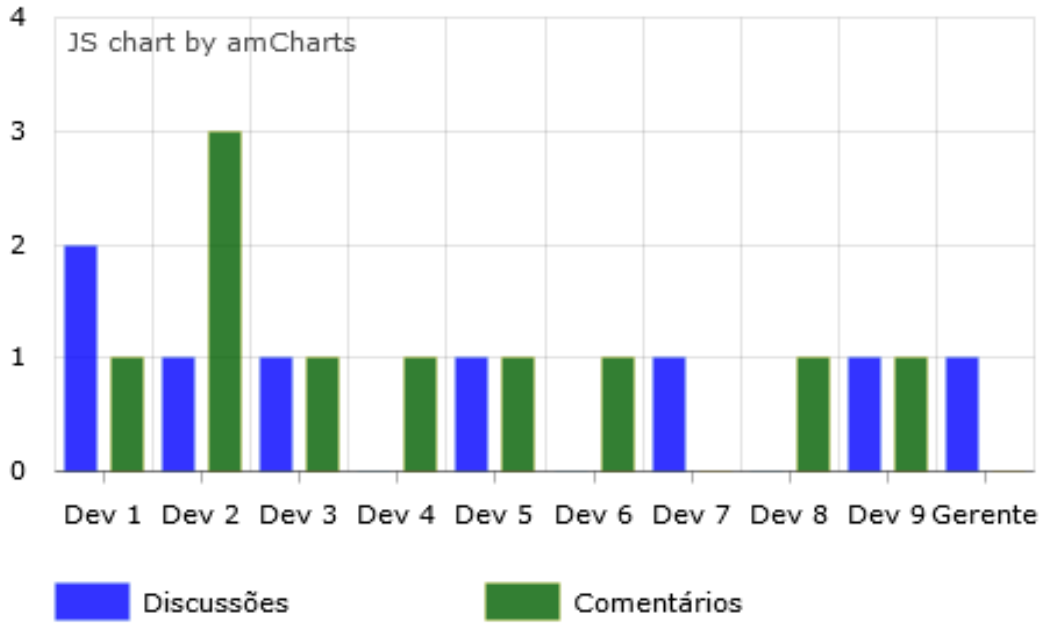

Figura 3.10: Activity Last Month View

No contexto de desenvolvimento distribuído a comunicação é considerado um fator importante para o sucesso de um projeto (PRESSMAN, 2005). Na seção seguinte é apresentado o módulo de comunicação desenvolvido com o objetivo ajudar na comunicação entre os membros da equipe permitindo uma maior interação.

3.7.6 Módulo comunicação

A figura 3.11 representa o módulo de comunicação que consiste em um fórum de discussões onde estão presentes todas as discussões iniciadas por qualquer entidade do sistema juntamente com os comentários realizados nas mesmas. Através deste fórum uma entidade pode iniciar uma nova discussão ou então comentar uma discussão iniciada previamente. Para iniciar uma discussão uma entidade deve identificar-se através da caixa de seleção que se encontra acima da caixa de texto onde será inserido o texto. Para inserir um comentário basta clicar no botão comentar referente a discussão escolhida que um campo texto será exibido junto com uma caixa de seleção para que a entidade possa identificar-se.

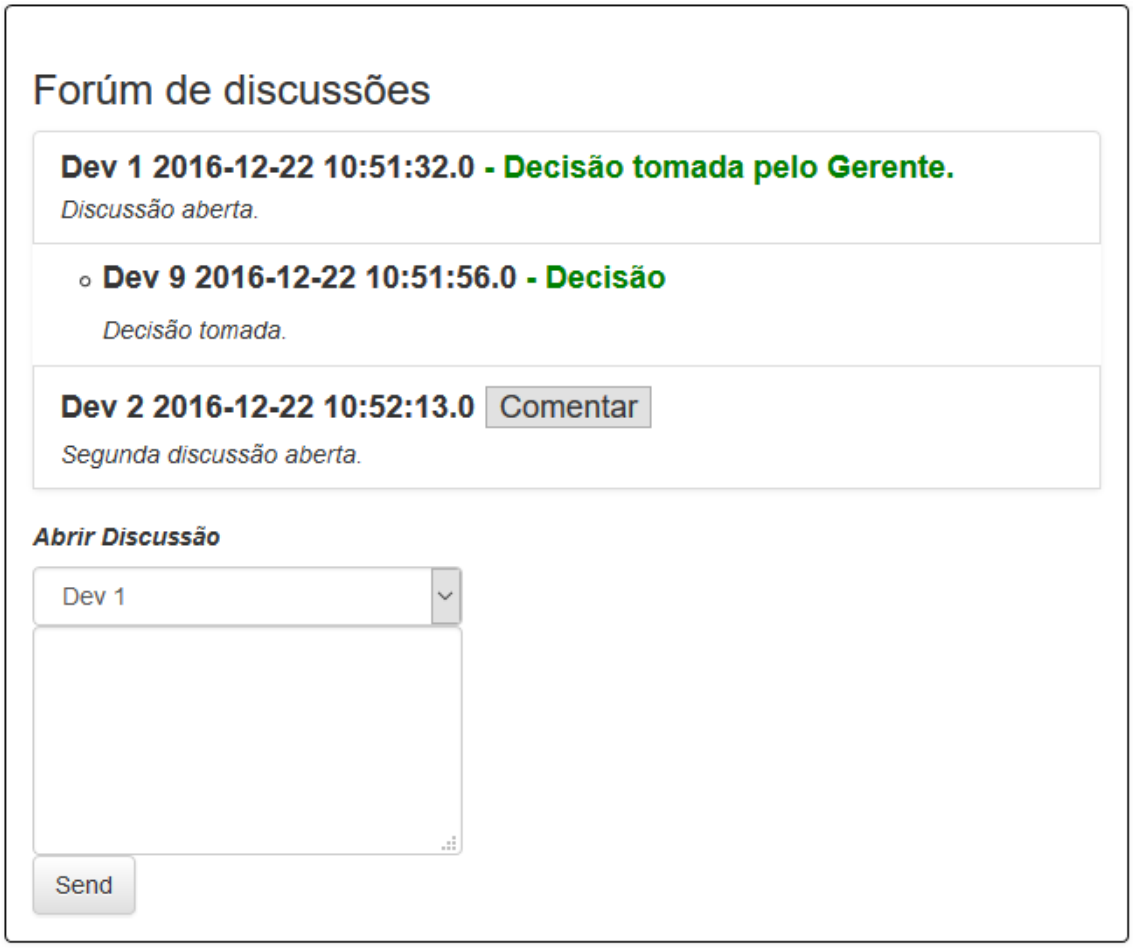

Figura 3.11: Fórum de discussões

Ainda através desse fórum, um membro de uma equipe responsável pelo processo de tomada de decisão, pode apoiar-se nas discussões e tomar sua decisão indicando quem tomou a decisão bem como indicando qual foi a decisão tomada.

O módulo de comunicação está presente na tela de detalhes de cada uma das visualizações previamente apresentadas. Nessa *view* de detalhes, cada discussão ou comentário realizado por uma entidade será vinculada a essa visualização.Dessa forma tanto as discussões quanto as decisões tomadas pelos membros de um equipe estarão no contexto dessa visualização.

Esse módulo também se encontra no painel geral de visualizações que é constituído por todas as visualizações da ArchiRiXCom. Nesse painel são listadas todas as discussões iniciadas juntamentes com os seus comentários associados informando quais discussões levaram a tomada de decisão e indicando quem foi o responsável por tomar a decisão.

3.7.7 Ferramentas

A implementação da solução foi realizada utilizando a linguagem de programação Java[2] através da IDE Eclipse[3]. Para persistência dos dados foi utilizada um banco de dados MySql[4] juntamente com a ferramenta phpMyAdmin[5] para administração do banco de dados.

Para a construção do modelo de entidades e relacionamentos foi utilizada a ferramenta MySql Workbench[6]. Para a construção do diagrama de casos de uso foi utilizada a ferramenta Astah[7].

Para a construção das visualizações e do *layout* foram utilizados os plugins: bootstrap[8], amcharts[9] e jQuery[10].

A solução foi desenvolvida obedecendo o padrão de projeto Model-View-Controller isolando cada uma das camadas da aplicação juntamente com o padrão DAO que consiste em isolar a camada de persistência de dados da camada de aplicação.

A construção das visualizações é realizada a partir de requisições HTTP GET que retorna um arquivo no formato JSON contendo os dados para a construção.

Nesse capítulo foi apresentada uma análise da arquitetura ArchiRi (LÉLIS et al., 2016) que serviu de base para o desenvolvimento da solução proposta no presente trabalho. A partir dessa análise foi proposta uma expansão para ArchiRiXCom que consiste em integrar novas visualizações sobre dados de reputação de desenvolvedores aumentando o poder de apoio a tomada de decisão da arquitetura. Junto a essas novas visualizações foi proposta a implementação de um módulo de comunicação que permita a interação entre os membros de uma equipe. Esse módulo foi desenvolvido no formato de um fórum de discussões permitindo aos usuários abrirem novas discussões, inserir comentários em discussões previamente abertas e tomarem decisões em cima de discussões. Também foram propostas duas outras visualizações sendo elas: *Activity View* e a *Activity Last Month*

[2]https://www.java.com/
[3]https://eclipse.org/
[4]https://www.mysql.com/
[5]https://www.phpmyadmin.net/
[6]http://www.mysql.com/products/workbench/
[7]http://astah.net/
[8]http://getbootstrap.com/
[9]https://www.amcharts.com/
[10]https://jquery.com/

View. Essas duas visualizações apresentam dados quanto a atividade dos membros de uma equipe no fórum de discussões. Ainda nesse capítulo foram apresentados os requisitos funcionais e não funcionais definidos para que a expansão fosse desenvolvida juntamente com o modelo de arquitetura e banco de dados, as novas visualizações, o módulo de comunicação e o ambiente de desenvolvimento.

No capítulo 4 será apresentada uma avaliação da ArchiRiXCom com o objetivo de mostrar como a arquitetura pode atuar no processo de apoio a tomada de decisão quanto a estimativas de esforço.

4 Análise da arquitetura ArchiRiXCom

Nesse capítulo será discutido como a ArchiRiXCom pode oferecer suporte à tomada de decisão com base em dados de reputação e comunicação entre membros de equipes. Para tal, utilizou-se um cenário onde o gerente de uma equipe precisa tomar uma decisão quanto a estimativa de esforço referente a uma requisição de mudança solicitada por um cliente. Tomou-se por base que o gerente estava de posse das estimativas realizadas por cada um dos membros envolvidos.

Para realizar essa análise foi realizada uma simulação onde a equipe é constituída por nove membros e até o momento essa equipe recebeu mil solicitações de requisição de mudança.

Cenário - Uma equipe de desenvolvimento de software que atua de maneira distribuída recebe uma demanda de uma requisição de mudança de um dos projetos em que está trabalhando. Ao receber a demanda, cada um dos desenvolvedores da equipe realiza uma estimativa de esforço informando quanto tempo acreditam que precisam para atendê-la. Após realizada a estimativa cada um dos desenvolvedores a envia ao gerente encarregado pela equipe . De posse dessas estimativas, o gerente precisa tomar a decisão sobre qual estimativa irá utilizar para essa demanda.

Uso da ArchiRiXCom - O primeiro passo do gerente ao acessar a ArchiRiXCom é procurar por dados de reputação dos seus desenvolvedores buscando aqueles que possuem uma maior reputação. A figura 4.1 mostra a *Ranking View* contendo dados de reputação de todos os desenvolvedores.

A partir dessa visualização o gerente pode perceber que os desenvolvedores com maior reputação são os desenvolvedores Dev 5 e Dev 1 com um valor de reputação de 0.1238 , seguidos do desenvolvedor Dev 2 com um valor de reputação de 0.1198 e pelo desenvolvedor Dev 9 com um valor de reputação de 0.1158.

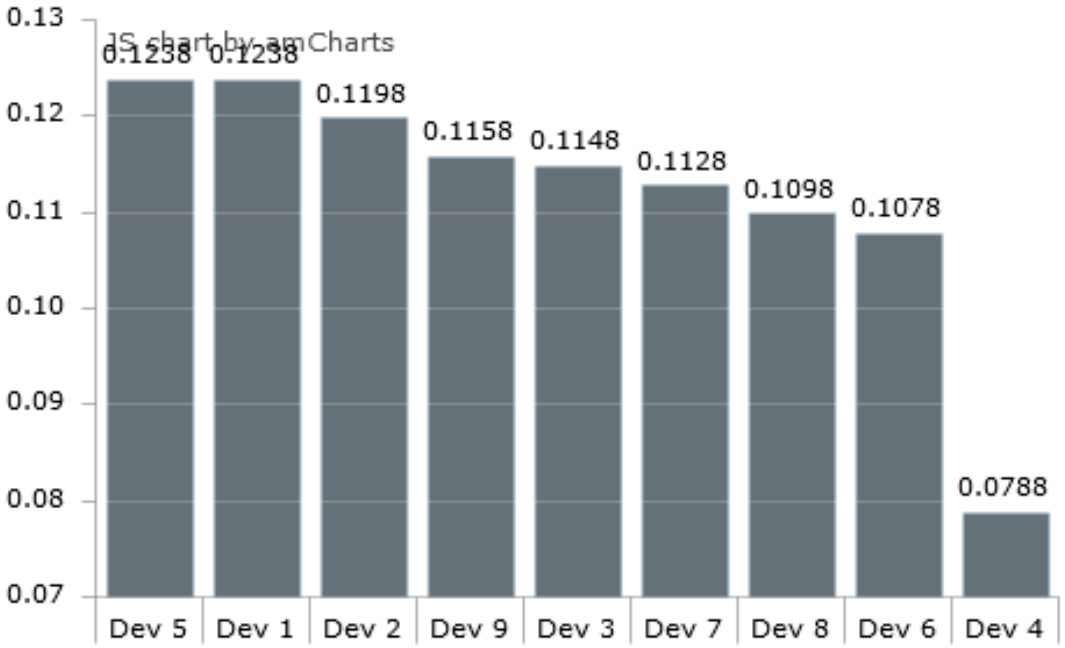

Figura 4.1: Cenário de uso - Ranking View

Feita essa análise o gerente irá avaliar a evolução da reputação dos seus desenvolvedores podendo analisar a mudança no comportamento de cada um deles. Para tal, o gerente busca a visualização *Historic View* e utiliza o filtro para exibir apenas os dados dos desenvolvedores selecionados na primeira análise.

A figura 4.2 apresenta a *Historic View* filtrada para exibir os dados dos desenvolvedores Dev 1, Dev 2, Dev 5 e Dev 9.

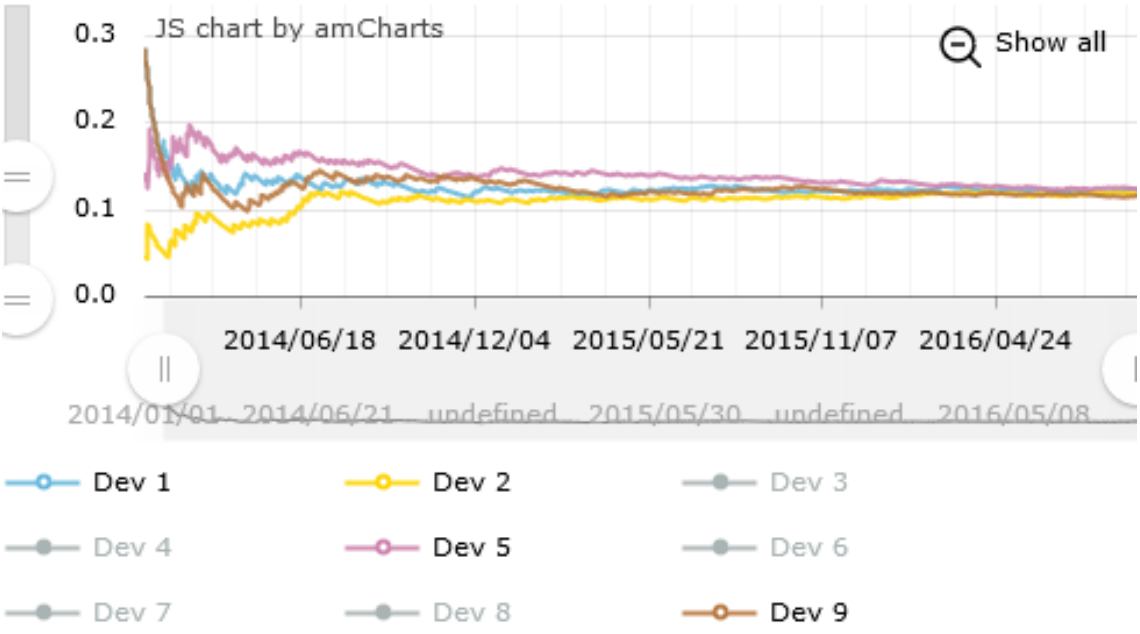

Figura 4.2: Cenário de uso - Historic View

Ao acessar esta visualização e filtrá-la, o gerente pode realizar uma análise mais precisa quanto a mudança de comportamento dos desenvolvedores bem como analisar como a reputação dos mesmo evoluiu.

Com base no modelo de cálculo de reputação, o gerente irá acessar a visualização *Criteria View* para poder entender como esses valores de reputação foram definidos e ainda analisar quais dos seus desenvolvedores receberam mais atribuições de requisições de mudança.

A figura 4.3 apresenta a *Criteria View* onde são informados o número de requisições de mudanças onde cada desenvolvedor recebeu um *"yes"*, ou seja, o número de requisições que foram atribuídas a ele, e o número de requisições de mudança onde cada desenvolvedor recebeu um *"no"*, ou seja, o número de requisições de mudança que não foram atribuídas a ele.

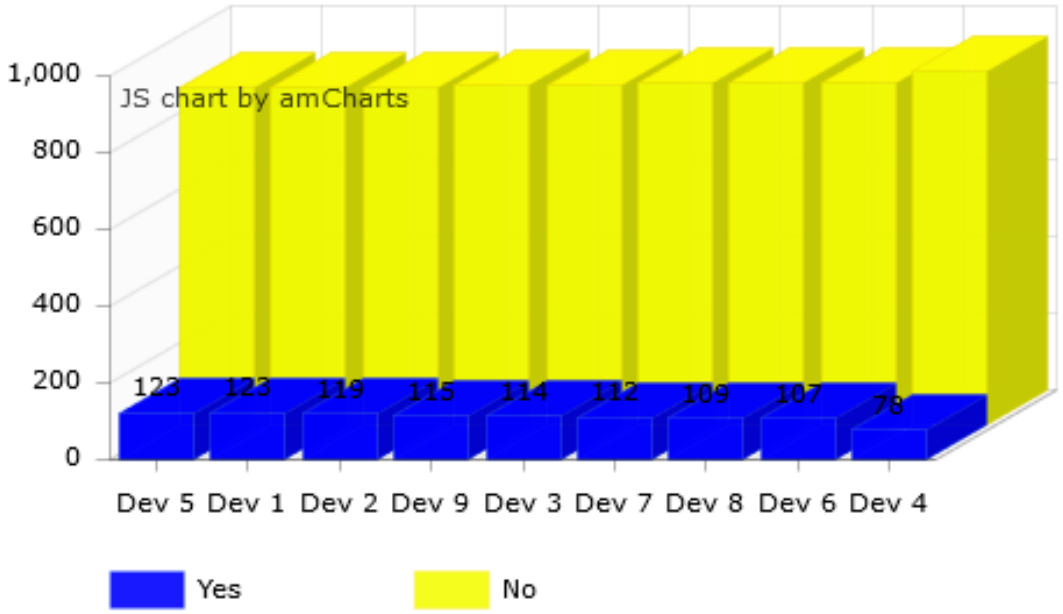

Figura 4.3: Cenário de uso - Criteria View

Através da *Criteria View* o gerente pode perceber que os desenvolvedores que receberam mais atribuições de requisições de mudança foram Dev 5 e Dev 1 empatados com 123 atribuições seguidos pelos desenvolvedores Dev 2 com 119 atribuições e Dev 9 com 115 atribuições.

Outro fator que o gerente pode avaliar para basear sua tomada de decisão é o grau de atividade dos desenvolvedores. Através dessa análise, pode verificar o comprometimento dos mesmos bem como o interesse em participar do processo de tomada de decisões. As figuras 4.4 e 4.5 apresentam respectivamente a *Activity View* e a *Activity Last Month* onde são apresentados os dados referentes às atividades dos desenvolvedores no fórum de discussões da ArchiRiXCom.

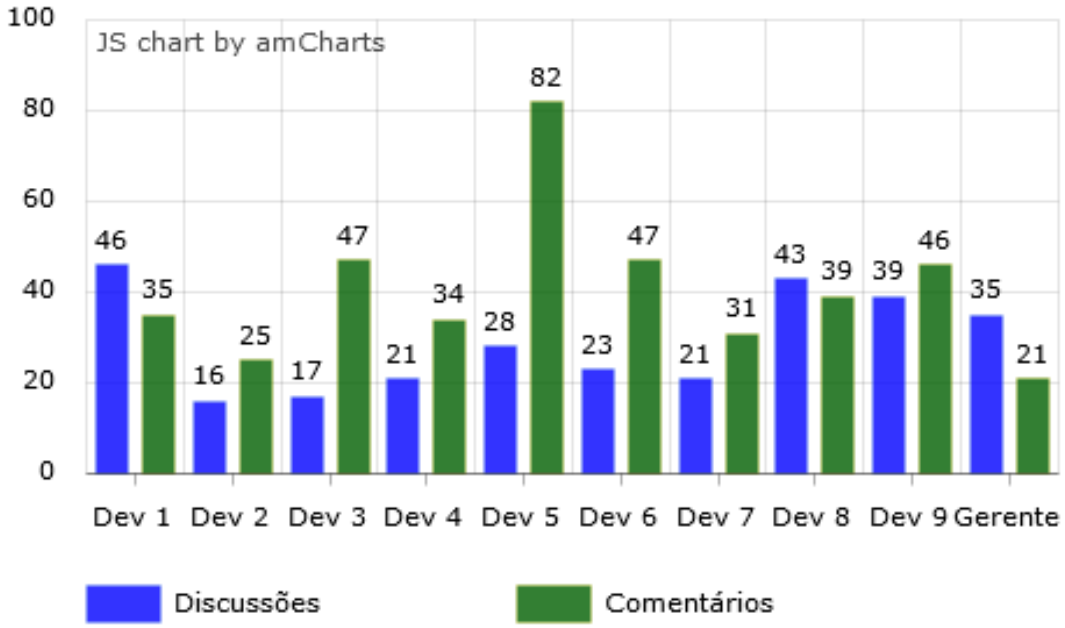

Figura 4.4: Cenário de uso - Activity View

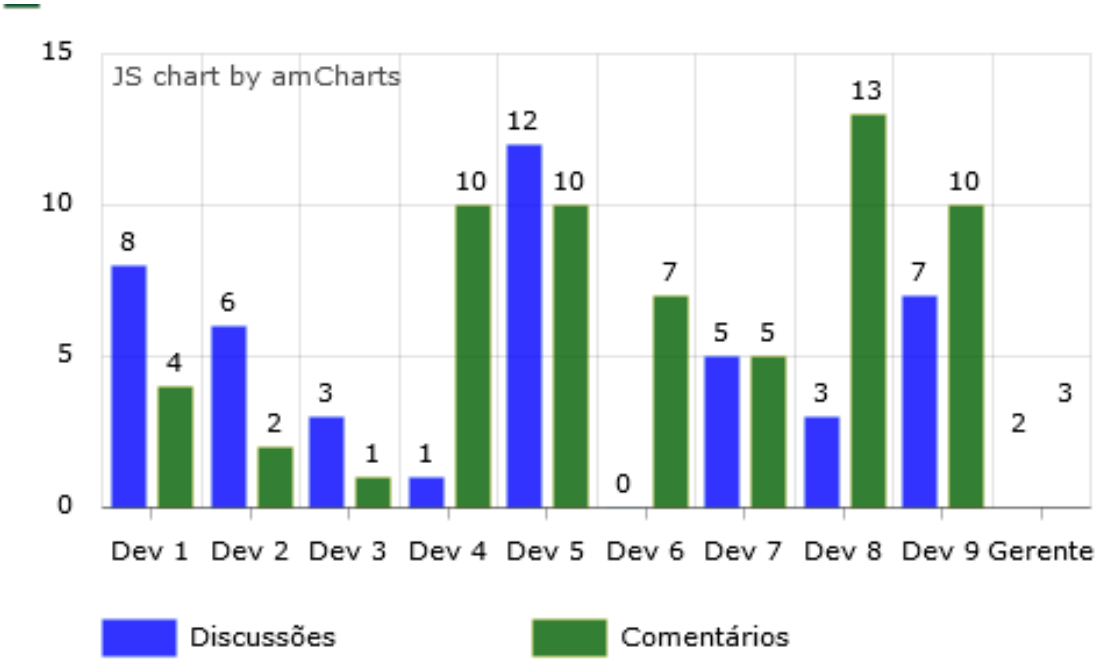

Figura 4.5: Cenário de uso - Activity Last Month View

A partir da figura 4.4 o gerente pode perceber que os desenvolvedores mais ativos no fórum de discussão são o Dev 5 tendo aberto 28 discussões e realizado 82 comentários e o Dev 1 tendo aberto 46 discussões e realizado 35 comentários. Já a partir da figura 4.5, o gerente pode perceber que os desenvolvedores mais ativos nos últimos 30 dias no fórum

de discussão são também o Dev 5 tendo aberto 12 discussões e realizado 10 comentários e o Dev 8 tendo aberto 3 discussões e realizado 13 comentários nesse período.

Ainda como forma de apoio a decisão a ser tomada o gerente pode utilizar o fórum de discussões para entrar em contato com os outros membros da equipe e pedir suas opiniões.

A figura 4.6 mostra uma discussão aberta pelo gerente e que está disponível para todos os membros da equipe comentarem.

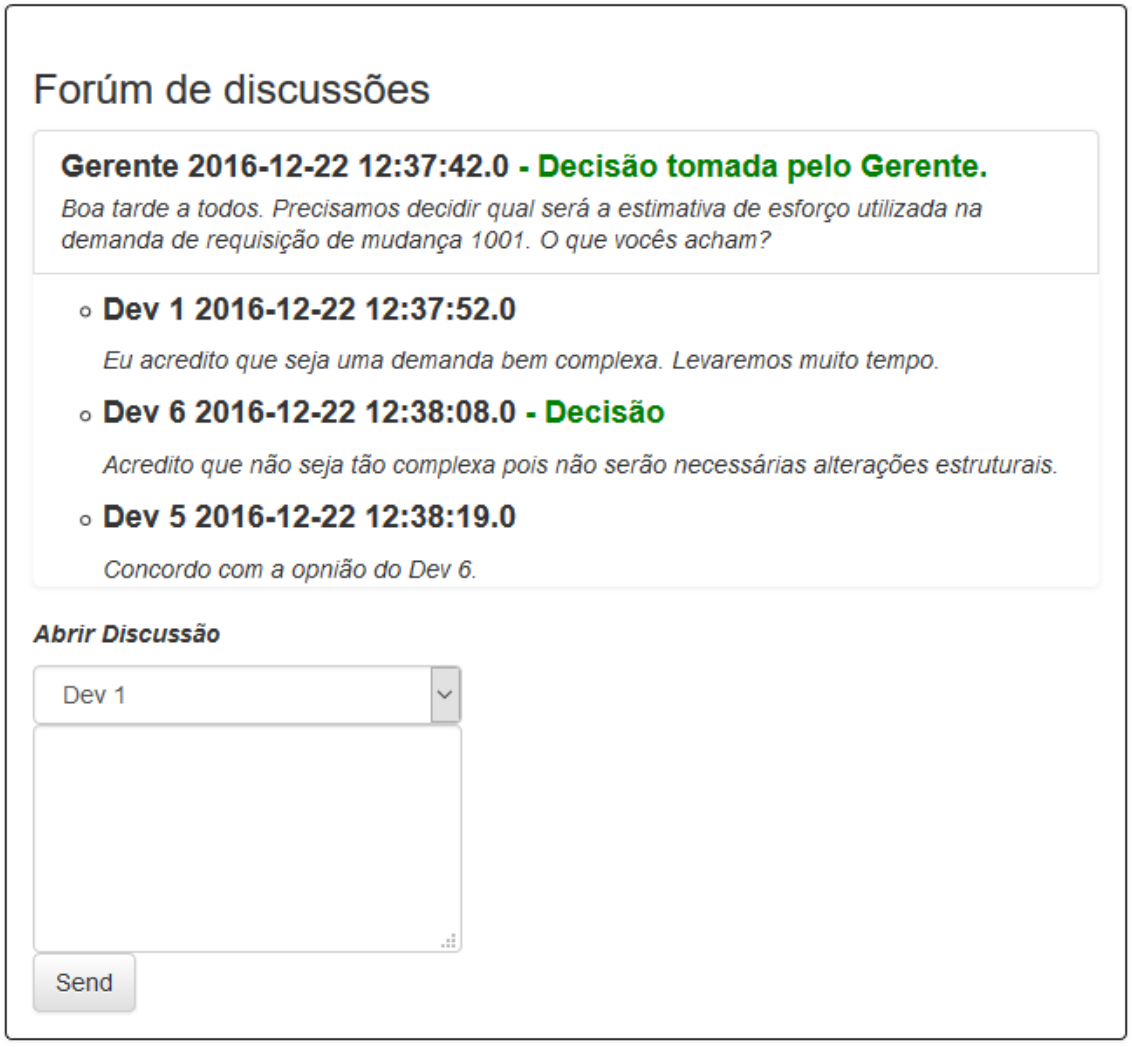

Figura 4.6: Cenário de uso - Fórum de discussões

Após realizada a discussão e alguns desenvolvedores opinarem sobre a dúvida levanta pelo gerente, o gerente decidiu a discussão com base no comentário do Dev 6. A partir do momento que a decisão é tomada a discussão é encerrada e nenhum membro da equipe poderá inserir comentários.

Este cenário mostrou como a ArchiRiXCom pode apoiar gerentes na análise sobre a reputação dos membros da sua equipe através das visualizações *Historic View, Ranking View e Criteria View* dessa forma apoiando a sua tomada de decisão.

Ainda foi apresentado como as visualizações *Activity View e Activity Last Month View* pode ajudar o gerente a analisar o interesse e comprometimento dos membros da sua equipe nas discussões que levam a tomadas de decisões.

Por fim, esse cenário mostrou como o módulo de comunicação pode contribuir para realização de discussões de forma a esclarecer dúvidas e dessa forma enriquecer as decisões tomadas.

5 Considerações finais

A reputação de desenvolvedores tem se tornado um ativo cada vez mais importante para as empresas que atuam na produção de software. No decorrer da etapa de pesquisa para a elaboração do presente trabalho, percebeu-se que no contexto de desenvolvimento de software distribuído, esta reputação têm sido um fator que pode impactar na colaboração e no relacionamento entre desenvolvedores que atuam em uma mesma equipe. A ferramenta ArchiRiXCom foi desenvolvida com o intuito de fornecer visualizações contendo dados de reputação para os membros de uma equipe de forma a possibilitar uma análise quanto a reputação dos mesmos.

No contexto de desenvolvimento de software distribuído, percebeu-se a importância da comunicação entre os membros envolvidos nesse processo, e que a distância entre os mesmos pode ser um complicador para essa atividade. Com base nisso, foi integrado a ArchiRiXCom um módulo de comunicação no formato de um fórum de discussões onde esses membros de equipe podem abrir discussões e comentar as discussões que foram abertas por outros membros. Através desse fórum os membros podem trocar ideias e apoiar a tomada de decisões.

Um fator importante nesse contexto é o nível de colaboração dos membros de uma equipe para com a equipe. Apesar de não ser abordado diretamente esse tópico, foram disponibilizadas duas visualizações na ArchiRiXCom onde pode-se avaliar quão ativo um membro da equipe é nos fóruns de discussão seja abrindo tópicos de discussão ou comentado as discussões abertas por outros membros.

Ao definir-se um modelo para o cálculo da reputação dos desenvolvedores, percebeu-se que existe uma gama de modelos para cálculo de reputação. A partir disso, a implementação da ferramenta foi modificada de forma a permitir que as visualizações propostas possam ser geradas a partir de diversos modelos diferentes.

A ferramenta ArchiRiXCom foi desenvolvida como uma expansão da arquitetura ArchiRi (LÉLIS et al., 2016). As novas visualizações geradas e o módulo de comunicação foram integrados as visualizações já existentes de forma a aumentar o poder de análise

sobre dados de reputação de ArchiRI.

A ferramenta ArchiRiXCom buscou apoiar às atividades de tomada de decisão referentes à estimativas de esforço. Uma limitação da ferramenta é o fato do apoio a tomada de decisão se basear apenas em dados de reputação e a troca de opiniões através do módulo de comunicação. Uma abordagem que abrange-se a forma como chegou-se às estimativas de esforço poderia enriquecer o processo de tomada de decisão.

Como trabalho futuro pretende-se expandir o módulo de comunicação gerado pela ArchiRiXCom. Hoje esse módulo é constituído apenas por um fórum de discussões. Realizar a expansão desse módulo com a adição de uma ferramenta de bate papo,por exemplo, pode enriquecer e apoiar as atividades de comunicação.

Outra possibilidade de trabalho futuro seria a implementação de um sistema baseado em níveis de permissão na ArchiRiXCom. No cenário atual, todos os membros de uma equipe possuem acesso às mesmas visualizações e possuem os mesmos privilégios. Fornecer um mecanismo de níveis de permissão pode otimizar às atividades de comunicação, uma vez que as discussões só poderiam ser visualizadas por que possuir o nível de permissão necessário bem como pode trazer uma segurança sobre dados sensíveis que possam vir a integrar a ferramenta.

Bibliografia

AL-ANI, B.; REDMILES, D. In strangers we trust? findings of an empirical study of distributed teams. In: IEEE. *2009 Fourth IEEE International Conference on Global Software Engineering*. [S.l.], 2009. p. 121–130.

BOSU, A.; CARVER, J. C. Impact of developer reputation on code review outcomes in oss projects: An empirical investigation. In: ACM. *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. [S.l.], 2014. p. 33.

CASARE, S.; SICHMAN, J. S. Using a functional ontology of reputation to interoperate different agent reputation models. *Journal of the Brazilian Computer Society*, SciELO Brasil, v. 11, n. 2, p. 79–94, 2005.

EVARISTO, J. R.; SCUDDER, R. Geographically distributed project teams: a dimensional analysis. In: IEEE. *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. [S.l.], 2000. p. 11–pp.

HINDS, P. *Distributed work*. [S.l.]: MIT Press, 2002.

ISENBERG, P. et al. Collaborative visualization: definition, challenges, and research agenda. *Information Visualization*, Sage Publications, v. 10, n. 4, p. 310–326, 2011.

JALALI, S.; WOHLIN, C. Agile practices in global software engineering-a systematic map. In: IEEE. *2010 5th IEEE International Conference on Global Software Engineering*. [S.l.], 2010. p. 45–54.

JØSANG, A. Trust and reputation systems. In: *Foundations of security analysis and design IV*. [S.l.]: Springer, 2007. p. 209–245.

JOSANG, A.; HALLER, J. Dirichlet reputation systems. In: IEEE. *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*. [S.l.], 2007. p. 112–119.

JØSANG, A.; QUATTROCIOCCHI, W. Advanced features in bayesian reputation systems. In: SPRINGER. *International Conference on Trust, Privacy and Security in Digital Business*. [S.l.], 2009. p. 105–114.

KIM, S.; JR, E. J. W. How long did it take to fix bugs? In: ACM. *Proceedings of the 2006 international workshop on Mining software repositories*. [S.l.], 2006. p. 173–174.

KIM, S. et al. Hybrid wom collection and visualization method for reputation rating in online community. *Indian Journal of Science and Technology*, v. 8, n. 18, 2015.

LÉLIS, C. A. S. et al. ArchiRI - uma arquitetura baseada em ontologias para a troca de informações de reputação Alternative Title : ArchiRI - An ontology-based architecture for the exchange of reputation information. *XII Brazilian Symposium on Information Systems*, p. 60–67, 2016.

LIPNACK, J.; STAMPS, J. *Virtual teams: Reaching across space, time, and organizations with technology*. [S.l.]: Jeffrey Stamps, 1997.

MIGUEL, M. A. et al. Um framework para apoiar estimativa de esforço em atividades de manutenção e evolução de software. 2016.

MOCKUS, A.; HERBSLEB, J. Challenges of global software development. In: IEEE. *Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International*. [S.l.], 2001. p. 182–184.

NASCIMENTO, D. C. M.; DAVID, J. M. N.; CARNEIRO, G. F. *Apoiando a reputação em programação em par distribuída*. 2013.

NASCIMENTO, L. de C.; SANTORO, F. M. Análise de interações nas comunidades virtuais de software livre. 2009.

NAYAK, M.; SUESAOWALUK, P. Risk factors that affect collaborative software development. In: IEEE. *Management of Innovation and Technology, 2008. ICMIT 2008. 4th IEEE International Conference on*. [S.l.], 2008. p. 1366–1371.

NOVAIS, R. L. et al. Software evolution visualization: A systematic mapping study. *Information and Software Technology*, Elsevier, v. 55, n. 11, p. 1860–1883, 2013.

O'BRIEN, J.; MARAKAS, G. Introduction to information systems, mcgraw-hill/irwin. *New York*, 2010.

PAREDES, J.; ANSLOW, C.; MAURER, F. Information visualization for agile software development. In: IEEE. *Software Visualization (VISSOFT), 2014 Second IEEE Working Conference on*. [S.l.], 2014. p. 157–166.

PEIXOTO, C. E. L.; AUDY, J. L. N.; PRIKLADNICKI, R. The importance of the use of an estimation process. In: ACM. *Proceedings of the 2010 ICSE Workshop on Software Development Governance*. [S.l.], 2010. p. 13–17.

PRESSMAN, R. S. *Software engineering: a practitioner's approach*. [S.l.]: Palgrave Macmillan, 2005.

PRIKLADNICKI, R.; AUDY, J. L. N.; EVARISTO, J. R. Distributed software development: Toward an understanding of the relationship between project team, users and customers. In: *ICEIS (3)*. [S.l.: s.n.], 2003. p. 417–423.

PRIKLADNICKI, R. et al. Desenvolvimento distribuído de software: um modelo de classificação dos níveis de dispersão dos stakeholders. In: *I Brazilian Symposium on Information Systems (SBSI 04)*. [S.l.: s.n.], 2004.

PRIKLADNICKI, R. et al. Munddos: um modelo de referência para desenvolvimento distribuído de software. Pontifícia Universidade Católica do Rio Grande do Sul, 2003.

RAMARAO, P. et al. Impact of bug reporter's reputation on bug-fix times. In: IEEE. *2016 International Conference on Information Systems Engineering (ICISE)*. [S.l.], 2016. p. 57–61.

SÄNGER, J.; PERNUL, G. Visualizing transaction context in trust and reputation systems. In: IEEE. *Availability, Reliability and Security (ARES), 2014 Ninth International Conference on*. [S.l.], 2014. p. 94–103.

TAVARES, J. F. et al. Giveme infra: Uma infraestrutura baseada em m ultiplas visões interativas para apoiar a evolu ção distribu ída de software. Universidade Federal de Juiz de Fora, 2015.

WEISS, C. et al. How long will it take to fix this bug? In: IEEE COMPUTER SOCIETY. *Proceedings of the Fourth International Workshop on Mining Software Repositories*. [S.l.], 2007. p. 1.

ZACHARIA, G.; MOUKAS, A.; MAES, P. Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, Elsevier, v. 29, n. 4, p. 371–388, 2000.