

<p>2.3.6 Ameças à Validade 48</p> <p>2.4 Resumo do Capítulo 49</p> <p>3 Mapeamento Sistemático da Literatura 51</p> <p>3.1 Introdução 51</p> <p>3.2 Metodologia de Pesquisa 52</p> <p>3.2.1 Questões de Pesquisa 52</p> <p>3.2.2 Pesquisa da Literatura 52</p> <p>3.2.3 Esquemas de Classificação 54</p> <p>3.3 Resultados 57</p> <p>3.3.1 Frequência das Publicações 57</p> <p>3.3.2 Classificação por Dimensões de Melhoria 57</p> <p>3.3.3 Suporte à Papéis da Manutenção de Software 63</p> <p>3.4 Discussão 65</p> <p>3.5 Limitações e Ameças à Validade 66</p> <p>3.6 Trabalhos Relacionados 67</p> <p>3.7 Resumo do Capítulo 68</p> <p>4 Levantamento por Questionário com Profissionais 69</p> <p>4.1 Introdução 69</p> <p>4.2 Objetivo do Levantamento com Profissionais 70</p> <p>4.3 Desenho e Metodologia da Pesquisa com Profissionais 71</p> <p>4.3.1 Conceitos Básicos 71</p> <p>4.3.2 Metodologia 72</p> <p>4.4 Resultados 78</p> <p>4.4.1 Perfil dos Participantes 78</p> <p>4.4.2 Nível de Satisfação com as FGRM 80</p> <p>4.4.3 Avaliação das Funcionalidades Existentes 82</p> <p>4.4.4 Práticas Ágeis na Manutenção de Software 84</p> <p>4.5 Discussão 85</p> <p>4.6 Ameças à Validade 86</p> <p>4.7 Resumo do Capítulo 90</p> <p>5 Sugestões de Melhorias para as FGRM's 91</p> <p>5.1 Introdução 91</p> <p>5.2 Sugestões de Melhorias para as FGRM's 92</p> <p>5.2.1 Suporte à Qualidade do Relato 92</p> <p>5.2.2 Busca por Código Fonte 93</p>	<p>2.3.6 Ameças à Validade 48</p> <p>2.4 Resumo do Capítulo 49</p> <p>3 Mapeamento Sistemático da Literatura 51</p> <p>3.1 Introdução 51</p> <p>3.2 Metodologia de Pesquisa 52</p> <p>3.2.1 Questões de Pesquisa 52</p> <p>3.2.2 Pesquisa da Literatura 52</p> <p>3.2.3 Esquemas de Classificação 54</p> <p>3.3 Resultados 57</p> <p>3.3.1 Frequência das Publicações 57</p> <p>3.3.2 Classificação por Dimensões de Melhoria 57</p> <p>3.3.3 Suporte à Papéis da Manutenção de Software 63</p> <p>3.4 Discussão 65</p> <p>3.5 Limitações e Ameças à Validade 66</p> <p>3.6 Trabalhos Relacionados 67</p> <p>3.7 Resumo do Capítulo 68</p> <p>4 Levantamento por Questionário com Profissionais 69</p> <p>4.1 Introdução 69</p> <p>4.2 Objetivo do Levantamento com Profissionais 70</p> <p>4.3 Desenho e Metodologia da Pesquisa com Profissionais 71</p> <p>4.3.1 Conceitos Básicos 71</p> <p>4.3.2 Metodologia 72</p> <p>4.4 Resultados 78</p> <p>4.4.1 Perfil dos Participantes 78</p> <p>4.4.2 Nível de Satisfação com as FGRM 80</p> <p>4.4.3 Avaliação das Funcionalidades Existentes 82</p> <p>4.4.4 Práticas Ágeis na Manutenção de Software 84</p> <p>4.5 Discussão 85</p> <p>4.6 Ameças à Validade 89</p> <p>4.7 Resumo do Capítulo 90</p> <p>5 Sugestões de Melhorias para as FGRM's 91</p> <p>5.1 Introdução 91</p> <p>5.2 Sugestões de Melhorias para as FGRM's 92</p> <p>5.2.1 Suporte à Qualidade do Relato 92</p> <p>5.2.2 Busca por Código Fonte 93</p>
--	--



Figura 2.5: Informações que compõem uma RM

Situação A situação atual de uma RM. Representa os diversos estados que uma RM possui em seu ciclo de vida. Nesta dissertação discutimos brevemente o ciclo de vida de uma RM na Subseção 2.1.3.2.

Criado Por Nome da pessoa ou um identificador já registrado no sistema de quem criou a RM.

Atribuído Para A RM pode ser atribuída a uma pessoa específica caso ela seja capaz de resolvê-la, caso contrário, a RM será atribuída para alguém que possuir o papel de definir o desenvolvedor mais apto para solucionar aquela RM. Neste estudo, o membro de uma equipe de manutenção com festa função é denominado *Agente de Triagem*.

Anexo Refere-se a informação que não esteja em formato de texto e que podem ser incluída na RM como casos de teste, capturas de tela, cadeia de registros de ativação (stack trace), dentre outros.

Comentário Registra o histórico de discussões realizadas durante o processo de resolução da RM.

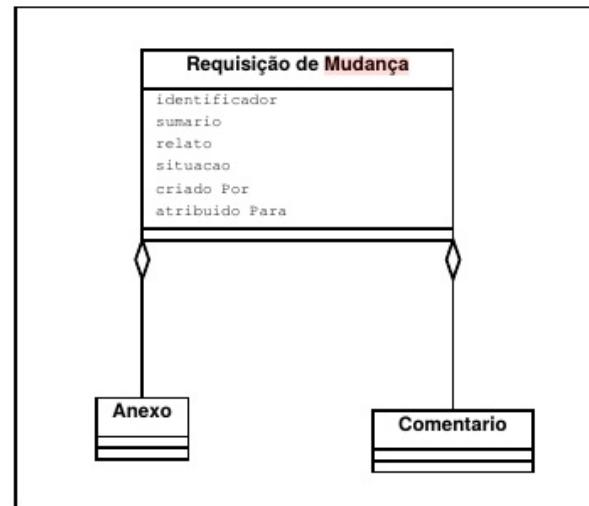


Figura 2.5: Informações que compõem uma RM

Situação A situação atual de uma RM. Representa os diversos estados que uma RM possui em seu ciclo de vida. Nesta dissertação discutimos brevemente o ciclo de vida de uma RM na Subseção 2.1.3.2.

Criado Por Nome da pessoa ou um identificador já registrado no sistema de quem criou a RM.

Atribuído Para A RM pode ser atribuída a uma pessoa específica caso ela seja capaz de resolvê-la, caso contrário, a RM será atribuída para alguém que possuir o papel de definir o desenvolvedor mais apto para solucionar aquela RM. Neste estudo, o membro de uma equipe de manutenção com festa função é denominado *Agente de Triagem*.

Anexo Refere-se a informação que não esteja em formato de texto e que podem ser incluída na RM como casos de teste, capturas de tela, cadeia de registros de ativação (stack trace), dentre outros.

Comentário Registra o histórico de discussões realizadas durante o processo de resolução da RM.

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

75

Questão 01 Qual a opinião dos profissionais envolvidos em Manutenção de Software com relação as funcionalidades oferecidas atualmente pelas FGRM?

Questão 02 Na visão dos profissionais envolvidos em Manutenção de Software quais das extensões propostas na literatura teriam maior relevância em suas atividades atuais?

Questão 03 Como as práticas propostas pelos agilistas estão sendo utilizadas especialmente no processo de manutenção de software?

Questão 04 Como as FGRMs podem ajudar aos times devotados à manutenção de software na prática adotada pelos agilistas?

O desenho da pesquisa é detalhado na próxima seção onde discutimos a estrutura do questionário bem como a amostra da população que foi utilizada.

4.3 Desenho e Metodologia da Pesquisa com Profissionais

4.3.1 Conceitos Básicos

Estudos primários em Engenharia de Software (SE), como os levantamentos por questionário, são muitas vezes conduzidos em amostras estabelecidas por conveniência [Sjøberg et al., 2005, Dybå et al., 2006]. Um desafio no estabelecimento de amostras representativas, especialmente em Engenharia de Software, é a identificação de fontes relevantes e disponíveis que permitam criar as amostragem [de Mello et al., 2014]. Uma alternativa é a utilização de fontes disponíveis na Internet, como as rede sociais, para aumentar o tamanho da amostra [de Mello & Travassos, 2013]. Outra fonte que pode aumentar a significância das amostras são os projetos de código e seus respectivos artefatos.

O nosso levantamento com profissionais consistiu de um estudo exploratório sem uma hipótese prévia. Idealmente, este levantamento deveria ser aplicado em todos os profissionais envolvidos em desenvolvimento e manutenção de software e que tenham feito o uso razoável de alguma FGRM. Naturalmente não é possível alcançar aquela população. Desta forma, foi utilizada uma estratégia de amostragem de *conveniência* onde o questionário foi aplicado em dois grupos distintos. Uma amostragem de *conivência* é técnica não probabilística de amostragem onde os sujeitos são seleciona-

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

75

Questão 01 Qual a opinião dos profissionais envolvidos em Manutenção de Software com relação as funcionalidades oferecidas atualmente pelas FGRM?

Questão 02 Na visão dos profissionais envolvidos em Manutenção de Software quais das extensões propostas na literatura teriam maior relevância em suas atividades atuais?

Questão 03 Como as práticas propostas pelos agilistas estão sendo utilizadas especialmente no processo de manutenção de software?

Questão 04 Como as FGRMs podem ajudar aos times devotados à manutenção de software na prática adotada pelos agilistas?

O desenho da pesquisa é detalhado na próxima seção onde discutimos a estrutura do questionário bem como a amostra da população que foi utilizada.

4.3 Desenho e Metodologia da Pesquisa com Profissionais

4.3.1 Conceitos Básicos

Estudos primários em Engenharia de Software (SE), como os levantamentos por questionário, são muitas vezes conduzidos em amostras estabelecidas por conveniência [Sjøberg et al., 2005, Dybå et al., 2006]. Um desafio no estabelecimento de amostras representativas, especialmente em Engenharia de Software, é a identificação de fontes relevantes e disponíveis que permitam criar as amostragem [de Mello et al., 2014]. Uma alternativa é a utilização de fontes disponíveis na Internet, como as rede sociais, para aumentar o tamanho da amostra [de Mello & Travassos, 2013]. Outra fonte que pode aumentar a significância das amostras são os projetos de código e seus respectivos artefatos.

O nosso levantamento com profissionais consistiu de um estudo exploratório sem uma hipótese prévia. Idealmente, este levantamento deveria ser aplicado em todos os profissionais envolvidos em desenvolvimento e manutenção de software e que tenham feito o uso razoável de alguma FGRM. Naturalmente não é possível alcançar aquela população. Desta forma, foi utilizada uma estratégia de amostragem de *conveniência* onde o questionário foi aplicado em dois grupos distintos. Neste tipo de amostragem a seleção de indivíduos é realizada por conta de sua facilidade de acesso

dos por conta de sua facilidade de acesso ou proximidade [Etikan et al., 2016]. Maiores detalhes sobre o processo de amostragem serão discutidos na Seção 4.3.2.1.

De modo a dar validade as amostras selecionadas para este estudo utilizamos o arcabouço (framework) conceitual proposto por de Mello e outros [de Mello et al., 2014]. O modelo proposto inclui além dos conceitos estatísticos tradicionalmente utilizados em levantamentos com questionário, tais como público-alvo, população, amostragem e unidade de observação [Thompson, 2012], discute conceitos relacionados com *Fonte de Amostragem, Unidade de Pesquisa, Plano de Pesquisa e Estratégia de Amostragem*.

Uma *Fonte de Amostragem* consiste de um banco de dados, que não necessariamente é automatizado, em que um subconjunto válido do público-alvo pode ser sistematicamente recuperado, além de permitir a extração aleatória de amostras da população de interesse. Sendo assim, os autores [de Mello et al., 2014] afirmam que no caso de determinada *Fonte de Amostragem* ser considerada válida para um contexto de pesquisa específico, pode-se concluir que as amostras podem ser extraídas desta fonte a fim de serem utilizados no mesmo contexto de pesquisa. Para ser considerada válida, uma *Fonte de Amostragem* deve satisfazer, pelo menos, aos seguintes Requisitos Essenciais (Essential Requirements- ER):

4.3.2 Metodologia

No caso deste levantamento por questionário, o público-alvo é o conjunto de profissionais que trabalham com manutenção de software e que tenham uma razoável experiência de uso das FGRMs. Naturalmente o perfil da população de interesse é bastante geral, uma vez que as características e práticas do profissional é bastante diverso e pode depender de questões como processo de software e linguagem de programação utilizada, tipo de projeto no qual está envolvido, dentre outras. Assim, todos os desenvolvedores de código aberto e código proprietário, que de alguma forma tenham utilizado determinada FGRM, podem potencialmente contribuir com esta investigação. É importante destacar que coube ao Plano de Pesquisa avaliar sobre a inclusão de determinado participante utilizando, por exemplo, o nível de experiência do respondente.

4.3.2.1 Fontes de Amostragem

Uma *Fonte de Amostragem* consiste de um banco de dados, que não necessariamente é automatizado, em que um subconjunto válido da população pode ser sistematicamente recuperado, além de permitir a extração aleatória de amostras da população de interesse [de Mello et al., 2014]. Utilizamos neste estudo as duas Fontes de Amostragem exibidas na Tabela ???. Na primeira fonte temos, indivíduos relacionados a grupos que

ou proximidade [Marshall, 1996]. Maiores detalhes sobre o processo de amostragem serão discutidos na Seção 4.3.2.1.

O trabalho de Mello e outros [de Mello et al., 2014] apresenta um arcabouço (framework) conceitual para a determinação de fontes adequadas para amostragens de profissionais em levantamentos na área de Engenharia de Software. Decidimos utilizar alguns aspectos deste arcabouço para discutir a adequação da nossa amostra. O modelo proposto inclui além dos conceitos estatísticos tradicionalmente utilizados em levantamentos com questionário, tais como público-alvo, população, amostragem e unidade de observação [Thompson, 2012], discute conceitos relacionados com *Fonte de Amostragem, Unidade de Pesquisa, Plano de Pesquisa e Estratégia de Amostragem*.

No trabalho de De Mello e outros [de Mello et al., 2014] afirma-se que uma *Fonte de Amostragem* deve ser organizada utilizando um sistema de banco de dados possibilitando a extração de subconjuntos da amostra disponível da população. Sendo assim, os autores discutem que no caso de uma *Fonte de Amostragem* ser considerada válida para um contexto de pesquisa específico, pode-se concluir que as amostras podem ser extraídas desta fonte também podem ser consideradas válidas.

4.3.2 Metodologia

No caso deste levantamento por questionário, o público-alvo é o conjunto de profissionais que trabalham com manutenção de software e que tenham uma razoável experiência de uso das FGRMs. A caracterização e estratificação da população que temos interesse não é simples. Neste sentido, é difícil dizer que um extrato com uma certa experiência com FGRMs é mais relevante do que outro com maior tempo de uso deste tipo de ferramenta ou ainda questões como processo de software ou linguagem de programação. Salvo melhor juízo, todos os desenvolvedores de código aberto e código proprietário, que de alguma forma tenham utilizado alguma FGRM, podem ser relevantes nesta investigação. Elaboramos um Plano de Pesquisa onde excluímos participantes conforme critérios que serão detalhados a seguir.

4.3.2.1 Fontes de Amostragem

Uma *Fonte de Amostragem* consiste de um banco de dados, que não necessariamente é automatizado, em que um subconjunto válido da população pode ser sistematicamente recuperado, além de permitir a extração aleatória de amostras da população de interesse [de Mello et al., 2014]. Utilizamos neste estudo as duas Fontes de Amostragem exibidas na Tabela ???. Na primeira fonte, temos a expectativa de encontrarmos indivíduos ligados ao desenvolvimento da plataforma Python correspondam ou representem

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

77

contribuem para projetos de código aberto. Na segunda, esperamos encontrar profissionais que façam parte de empresas privadas de desenvolvimento³ e manutenção de software (Stack Overflow).

Identificador	Fonte de Amostragem	URL	Membros
FA01	Python	https://bugs.python.org/	~19 K
FA02	Stack Overflow	https://stackoverflow.com	~6 M

Tabela 4.1: Fontes de Amostragem utilizadas no levantamento com questionário.

A fonte FA01 foi utilizada por apresentar as seguintes características: (i) pelo menos 5 anos de existência; (ii) comunidade bem estabelecida, no sentido de um número relevante e participativo de contribuidores e usuários; (iii) permite acesso aos dados históricos de suas RMAs. Por outro lado, a fonte FA02 foi selecionada por devido à sua cobertura, que conta com mais de 6 milhões de usuários⁴.

Conforme anteriormente discutido, este levantamento fez uso de uma amostragem de conveniência. Embora este tipo de estratégia de amostragem apresentar limitações devido a natureza subjetiva na escolha da amostra, ela é útil especialmente quando randomização não é possível, como no caso de uma população muito grande [Etikan et al., 2016]. No estudo conduzido por de Melo e outros [?] é apresentam quatro itens denominados “Requisitos Essenciais” identificados como ER1, ER2, ER3 e ER4. Estes itens dão validade a uma fonte de amostragem. Nossa avaliação é que nossa fonte satisfaz bem todos requisitos.

4.3.2.2 Construção das Fontes de Amostragem

A unidade básica de informação para construirmos as Fontes de Amostragem foram a lista de RMAs disponível em sua respectiva FGRM⁵ (FA01) e as discussões propostas pelos usuários (FA02). Em ambos os casos foram coletados os atributos:

- Nome do Participante
- E-mail do Participante
- Data de Ação

³O nosso foco é com profissionais envolvidos com manutenção de software, contudo, não seria possível separar os que trabalham com desenvolvimento daqueles que se dedicam a manutenção de software.

⁴Disponível em <http://stackexchange.com/sites>. Acessado em novembro de 2016.

⁵<http://bugs.python.com>

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

77

profissionais do extrato de código aberto. A segunda fonte, FA02, corresponde a indivíduos com interesse na rede social denominada Stack Overflow e neste caso devemos encontrar um perfil mais abrangente de desenvolvedores e mantenedores de software³.

Identificador	Fonte de Amostragem	URL	Membros
FA01	Python	https://bugs.python.org/	~19 K
FA02	Stack Overflow	https://stackoverflow.com	~6 M

Tabela 4.1: Fontes de Amostragem utilizadas no levantamento com questionário.

A fonte FA01 foi utilizada por apresentar as seguintes características: (i) pelo menos 5 anos de existência; (ii) comunidade bem estabelecida, no sentido de um número relevante e participativo de contribuidores e usuários; (iii) permite acesso aos dados históricos de suas RMAs. Por outro lado, a fonte FA02 foi selecionada por devido à sua cobertura, que conta com mais de 6 milhões de usuários⁴.

Conforme anteriormente discutido, este levantamento fez uso de uma amostragem de conveniência. Embora este tipo de estratégia de amostragem apresentar limitações devido a natureza subjetiva na escolha da amostra, ela é útil especialmente quando a randomização não é possível, como no caso de uma população muito grande ou de difícil caracterização [Boxill et al., 1997]. No estudo conduzido por de Melo e outros [de Melo et al., 2014] é apresentam quatro itens denominados “Requisitos Essenciais” identificados como ER1, ER2, ER3 e ER4. Estes itens dão validade a uma fonte de amostragem. Nossa avaliação é que nossa fonte satisfaz bem todos requisitos.

4.3.2.2 Construção das Fontes de Amostragem

A unidade básica de informação para construirmos as Fontes de Amostragem foram a lista de RMAs disponível em sua respectiva FGRM⁵ (FA01) e as discussões propostas pelos usuários (FA02). Em ambos os casos foram coletados os atributos:

- Nome do Participante
- E-mail do Participante
- Data de Ação
- Tipo de Ação

³Não colocamos esforço em tentar distinguir se o foco de atividade do usuário do Stack Overflow é desenvolvimento, manutenção ou outra categoria

⁴Disponível em <http://stackexchange.com/sites>. Acessado em novembro de 2016.

⁵<http://bugs.python.com>

- Tipo de Ação

O Tipo de Ação representa a aquilo que o participante realizou na Fonte de Amostragem, por exemplo relatar uma RM, finalizar uma RM, responder a uma pergunta e etc. Estes atributos foram utilizados para avaliar se determinada Unidade de Observação (indivíduo) seria incluído no quadro de amostragem, que é o conjunto final de potenciais participantes do estudo. Além destes atributos foram coletadas outras informações através do questionário de pesquisa (instrumento de medição) de modo a conhecer cada Unidade de Observação que participou do estudo tais como localização geográfica, tempo de experiência, nome da função desempenhada, principais atribuições, dentre outros.

No caso do Stack Overflow utilizamos uma métrica adicional da própria rede social conhecida como reputação⁶ que é uma medida aproximada de quanto a comunidade poderia confiar em determinado participante. A métrica é calculada com base nas ações do usuário e em como a comunidade avalia tais ações. Neste trabalho a ela foi utilizada para verificar a frequência de participação de determinado usuário em discussões sobre manutenção de software.

Para a extraímos os dados da rede social Stack Overflow utilizamos sua ferramenta web oficial que permite compartilhar, consultar e analisar os dados de todos os sites da rede Stack Exchange⁷. A ferramenta possibilita a utilização da linguagem SQL para acesso aos dados. A Figura [?]. É possível ainda extrair os dados formato CSV (Comma Separated Values) o qual foi posteriormente inserido em um banco de dados para aplicação das regras de inclusão e exclusão.

Para a fonte FA01 foi desenvolvido um Web Crawler para coletar as informações dos participantes. Um Web Crawler (rastreador web) é um programa de computador que navega pela World Wide Web de uma forma metódica e automatizada. A partir de uma lista de RMs previamente coletadas a ferramenta coletou os dados dos participantes a partir do histórico de modificações da mesma. A Figura [?] apresenta o histórico de registros de uma RM do projeto Python onde os dados dos participantes podem ser visualizados nos quadros inseridos. A ferramenta utiliza uma marcação HTML e os seu valor de classe (título, ou seja, nome de membro) para coletar os dados. Os dados coletados também foram armazenadas em um banco de dados para posterior aplicação de critérios de inclusão e exclusão.

⁶<http://stackoverflow.com/help/whats-reputation>

⁷<http://data.stackexchange.com/stackoverflow>

O Tipo de Ação representa a aquilo que o participante realizou na Fonte de Amostragem, por exemplo relatar uma RM, finalizar uma RM, responder a uma pergunta e etc. O tipo e a data da Ação foram utilizados para avaliar se o indivíduo estava no conjunto final de potenciais participantes do estudo. Além daqueles atributos foram coletadas outras informações através do questionário de pesquisa (instrumento de medição) de modo a conhecer cada profissional como por exemplo a localização geográfica, o tempo de experiência, o nome da função desempenhada, as principais atribuições, dentre outros.

No caso do Stack Overflow utilizamos uma métrica adicional da própria rede social conhecida como reputação⁶ que é uma medida aproximada de quanto a comunidade poderia confiar em determinado participante. A métrica é calculada com base nas ações do usuário e em como a comunidade avalia tais ações. Neste trabalho a ela foi utilizada para verificar a frequência de participação de determinado usuário em discussões sobre manutenção de software.

Para a extraímos os dados da rede social Stack Overflow utilizamos sua ferramenta web oficial que permite compartilhar, consultar e analisar os dados de todos os sites da rede Stack Exchange⁷. A ferramenta possibilita a utilização da linguagem SQL para acesso aos dados. A Figura 4.1 exibe a interface da ferramenta utilizada para coletados dos dados. É possível ainda extrair os dados formato CSV (Comma Separated Values) o qual foi posteriormente inserido em um banco de dados para aplicação das regras de inclusão e exclusão.

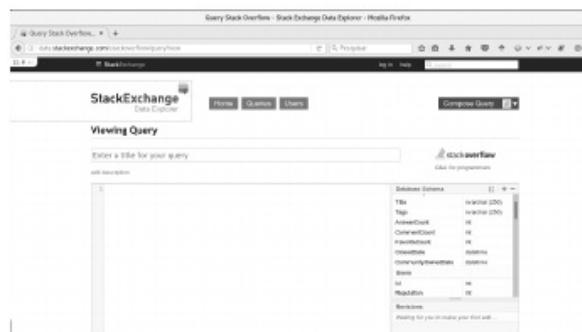


Figura 4.1: Ferramenta de coleta de dados da rede Stack Overflow

⁶<http://stackoverflow.com/help/whats-reputation>

⁷<http://data.stackexchange.com/stackoverflow>

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

79

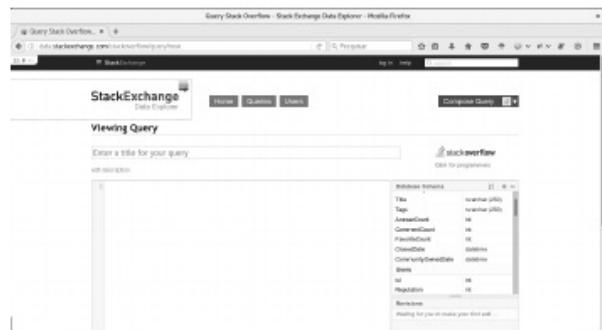


Figura 4.1: Ferramenta de coleta de dados da rede Stack Overflow

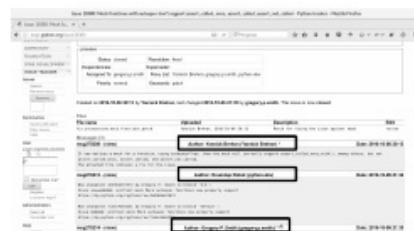


Figura 4.2: Histórico de relatos de uma RM do projeto Python

4.3.2.3 Seleção dos Participantes

Utilizamos estratégias distintas em cada fonte de amostragem para escolher os potenciais participantes do levantamento. Para a fonte FA01 utilizamos os registros históricos das RMs ocorridos nos últimos 05 anos. Além disso, foi coletada a frequência que um participante teve algum tipo de interação com projeto, como por exemplo abertura, solução ou comentários em RMs. Um participante seria incluído caso tivesse pelo menos uma interação no período avaliado.

No caso do Stack Overflow realizamos a busca de discussões que tinham relação com as sentenças de busca descritas na Figura 4.3. Um conjunto similar de sentenças de busca foi utilizado no mapeamento sistemático descrito no Capítulo 3. Para obtermos os dados utilizamos a busca oferecida pelo próprio site⁸. Neste contexto, visando restringir a seleção de grupos de participantes que estejam vinculados à desenvolvimento e manutenção de software aplicamos as seguintes regras de exclusão de participantes:

⁸<http://data.stackexchange.com/>

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

79

Para a fonte FA01 foi desenvolvido um Web Crawler para coletar as informações dos participantes. Um Web Crawler (rastreador web) é um programa de computador que navega pela World Wide Web de uma forma metódica e automatizada. A partir de uma lista de RMs previamente coletadas a ferramenta coletou os dados dos participantes a partir do histórico de modificações da mesma. A Figura 4.2 apresenta o histórico de registros de uma RM do projeto Python onde os dados dos participantes podem ser visualizados nos quadros inseridos. A ferramenta utiliza uma marcação HTML e os seu valor de classe (título, ou seja, nome de membro) para coletar os dados. Os dados coletados também foram armazenadas em um banco de dados para posterior aplicação de critérios de inclusão e exclusão.

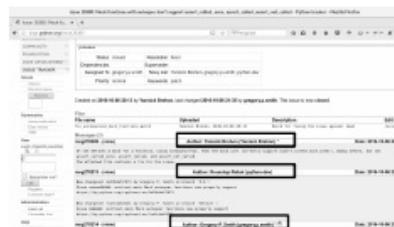


Figura 4.2: Histórico de relatos de uma RM do projeto Python

4.3.2.3 Seleção dos Participantes

Utilizamos estratégias distintas em cada fonte de amostragem para escolher os potenciais participantes do levantamento. Para a fonte FA01 utilizamos os registros históricos das RMs ocorridos nos últimos 05 anos. Além disso, foi coletada a frequência que um participante teve algum tipo de interação com projeto, como por exemplo abertura, solução ou comentários em RMs. Um participante seria incluído caso tivesse pelo menos uma interação no período avaliado.

No caso do Stack Overflow realizamos a busca de discussões que tinham relação com as sentenças de busca descritas na Figura 4.3. Um conjunto similar de sentenças de busca foi utilizado no mapeamento sistemático descrito no Capítulo 3. Para obtermos os dados utilizamos a busca oferecida pelo próprio site⁸. Neste contexto, visando restringir a seleção de grupos de participantes que estejam vinculados à desenvolvimento e manutenção de software aplicamos as seguintes regras de exclusão de participantes:

⁸<http://data.stackexchange.com/>

e manutenção de software aplicamos as seguintes regras de exclusão de participantes:

- Proibem expressamente a utilização dos seus dados, especialmente do seu endereço eletrônico, para a realização de estudos;
- A Fonte de Amostragem ao qual pertence não possui um mínimo de 05 anos de registros
- Para as discussões do Stack Overflow, aqueles que restringem explicitamente a mensagem individual entre seus membros;
- Utilizam uma língua diferente do inglês, tendo em vista que o idioma é padrão em fóruns internacionais e apenas existiam uma versão em inglês e português para o questionário utilizados.

```
("issue tracking" OR "bug tracking" OR
"issue-tracking" OR "bug-tracking" OR
"bug repository" OR "issue repository")
AND
("issue report" OR "bug report" OR
"bug prioritization" OR
"bug fix" OR "bug assignment" OR
"bug reassignment" OR "bug triage" OR
"duplicate bug" OR "reopened bug" OR
"bug impact" OR "bug localization" OR
"bug prediction" OR "bug risk" OR
"bugseverity" OR "bug classification")
```

Figura 4.3: Sentenças utilizadas para escolhas dos grupos do LinkedIn e de discussões no Stack Overflow

4.3.2.4 Questionário

O formulário enviado aos participantes foi estruturado em três **parte**, cada uma coletando um conjunto de informação. Na primeira parte estávamos interessados na formação de base (background) do respondente. O segundo conjunto de perguntas tinha por objetivo coletar a percepção dos participantes sobre as funcionalidades oferecidas pelas FGRMs. Na terceira parte estão as perguntas sobre as melhorias e proposição de novas funcionalidades para as FGRMs que foram propostas na literatura quando este trabalho foi realizado.

Antes de **aplicarmos** o formulário no público alvo foi realizado um processo de avaliação constituído de quatro etapas. O formulário resultante de uma etapa foi

- Proibem expressamente a utilização dos seus dados, especialmente do seu endereço eletrônico, para a realização de estudos;
- A Fonte de Amostragem ao qual pertence não possui um mínimo de 05 anos de registros
- Para as discussões do Stack Overflow, aqueles que restringem explicitamente a mensagem individual entre seus membros;
- Utilizam uma língua diferente do inglês, tendo em vista que o idioma é padrão em fóruns internacionais e apenas existiam uma versão em inglês e português para o questionário utilizados.

```
("issue tracking" OR "bug tracking" OR
"issue-tracking" OR "bug-tracking" OR
"bug repository" OR "issue repository")
AND
("issue report" OR "bug report" OR
"bug prioritization" OR
"bug fix" OR "bug assignment" OR
"bug reassignment" OR "bug triage" OR
"duplicate bug" OR "reopened bug" OR
"bug impact" OR "bug localization" OR
"bug prediction" OR "bug risk" OR
"bugseverity" OR "bug classification")
```

Figura 4.3: Sentenças utilizadas para escolhas dos grupos do LinkedIn e de discussões no Stack Overflow

4.3.2.4 Questionário

O formulário enviado aos participantes foi estruturado em três **parte**, cada uma coletando um conjunto de informação. Na primeira parte estávamos interessados na formação de base (background) do respondente. O segundo conjunto de perguntas tinha por objetivo coletar a percepção dos participantes sobre as funcionalidades oferecidas pelas FGRMs. Na terceira parte estão as perguntas sobre as melhorias e proposição de novas funcionalidades para as FGRMs que foram propostas na literatura quando este trabalho foi realizado.

Antes de **aplicarmos** o formulário no público alvo foi realizado um processo de avaliação constituído de quatro etapas. O formulário resultante de uma etapa foi **utilizado como entrada de outra imediatamente posterior**. As **etapas de avaliação** foram as seguintes:

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

81

utilizado como entrada de outra imediatamente posterior. As etapas de avaliação foram as seguintes:

- (i) Avaliação por Pesquisadores: Nesta etapa a primeira versão do formulário foi enviada para dois pesquisadores da área de manutenção de software.
- (ii) Avaliação por Profissionais: O formulário resultante da análise anterior foi encaminhado a dois profissionais que trabalham com manutenção de software.
- (iii) Piloto da Pesquisa: O formulário obtido da fase anterior foi utilizado em um piloto com dez profissionais envolvidos da manutenção de software de uma empresa pública de informática - PRODABEL⁹
- (iv) Tradução do Formulário: Em cada uma das etapas de anteriores o formulário foi aplicado em português, tendo em vista que alguns profissionais envolvidos no processo de avaliação não ter fluência em língua inglesa, em especial na fase ‘Piloto da Pesquisa’. Neste sentido, a última etapa consistiu na tradução do formulário para a língua inglesa. Esta etapa foi conduzida com o suporte de um pesquisador experiente na área de Engenharia de Software.

4.3.2.5 Envio da Mensagem

A fim de viabilizar e mitigar os riscos operacionais do envio manual de mensagens ao participantes foi desenvolvido um processo automatizado de remessa de mensagens aos participantes. O processo seguiu uma política que consiste em enviar uma mensagem ao participante com base em um modelo. Após um prazo de dois dias uma nova mensagem era enviada. Foi construída uma lista para incluir o endereço eletrônico daqueles que não gostariam de receber lembretes ou de participar da pesquisa de modo a respeitar a privacidade do participante. As mensagens foram preenchidas (uma a uma) e enviadas por meio de correio eletrônico com base no seguinte modelo:

⁹<http://www.prodabel.pbh.gov.br>

4.3. DESENHO E METODOLOGIA DA PESQUISA COM PROFISSIONAIS

81

- (i) Avaliação por Pesquisadores: Nesta etapa a primeira versão do formulário foi enviada para dois pesquisadores da área de manutenção de software.
- (ii) Avaliação por Profissionais: O formulário resultante da análise anterior foi encaminhado a dois profissionais que trabalham com manutenção de software.
- (iii) Piloto da Pesquisa: O formulário obtido da fase anterior foi utilizado em um piloto com dez profissionais envolvidos da manutenção de software de uma empresa pública de informática - PRODABEL⁹
- (iv) Tradução do Formulário: Em cada uma das etapas de anteriores o formulário foi aplicado em português, tendo em vista a falta de fluência em Inglês de alguns profissionais envolvidos no processo de avaliação, em especial na fase ‘Piloto da Pesquisa’. Neste sentido, a última etapa consistiu na tradução do formulário para a língua inglesa. Esta etapa foi conduzida com o suporte de um pesquisador experiente na área de Engenharia de Software.

4.3.2.5 Envio da Mensagem

A fim de viabilizar e mitigar os riscos operacionais do envio manual de mensagens aos participantes foi desenvolvido um processo automatizado de remessa de mensagens aos participantes. O processo seguiu uma política que consiste em enviar uma mensagem ao participante com base em um modelo. Após um prazo de dois dias uma nova mensagem era enviada. Foi construída uma lista para incluir o endereço eletrônico daqueles que não gostariam de receber lembretes ou de participar da pesquisa de modo a respeitar a privacidade do participante. As mensagens foram preenchidas (uma a uma) e enviadas por meio de correio eletrônico com base no seguinte modelo:

⁹<http://www.prodabel.pbh.gov.br>

Dear {{ nome do participante}}

I'm Vagner Clementino (homepages.dcc.ufmg.br/~vagnercs), Master Student at Federal University of Minas Gerais, Brazil. I'm conducting a research, supervised by ProfRodolfo Resende - homepages.dcc.ufmg.br/~rodolfo concerned with improvements in Issue Tracking System. As part of them, we planned and executed a survey aiming at to [reach](#) a large-scale population of researchers/practitioners interested on [to](#) improve the features of the Issue Tracking Systems. Based on your area of interest, we kindly invite you to take part in the following survey:

{{url do formulario}}

You [was](#) chosen because your relevant participation/contribution in {{nome da fonte de amostragem}}- {{url da fonte de amostragem}}. Your opinion is essential to strength our findings. Please, help us accordingly your possibilities by [answering](#) this survey until {{data limite}}. As soon as we conclude data analysis, we will share the results with all participants and the software engineering community. If you have already fulfilled this questionnaire, please ignore this email.

Thanks in advance,
Vagner Clementino

4.4 Resultados

Neste seção apresentamos os resultados obtidos da aplicação do questionário. Começamos com a análise do perfil dos respondentes. Em seguida, avaliamos o nível de satisfação que os participantes possuem com as ferramentas que eles utilizam. Posteriormente verificamos a adoção das metodologias propostas pelos agilistas no processo de desenvolvimento e em especial da manutenção de software.

4.4.1 Perfil dos Participantes

Antes de apresentarmos os resultado sobre as ferramentas, avaliamos o perfil dos respondentes. Como pode ser observado na Figura 4.4 a função mais frequente é a de desenvolvedor. Todavia, grande parte dos respondentes estão diretamente vinculados ao desenvolvimento e manutenção de software, tanto que mais de 80% da amostrada é formada por desenvolvedores, engenheiros de software, gerentes e arquitetos. Neste sentido, pelo menos no que tange à função desempenhada, obtivemos um conjunto significativo de participantes.

A distribuição geográfica dos participantes pode ser visualizada na Figura 4.5. Há

Dear {{ nome do participante}}

I'm Vagner Clementino (homepages.dcc.ufmg.br/~vagnercs), Master Student at Federal University of Minas Gerais, Brazil. I'm conducting a research, supervised by ProfRodolfo Resende - homepages.dcc.ufmg.br/~rodolfo concerned with improvements in Issue Tracking System. As part of them, we planned and executed a survey aiming at to [reaching](#) a large-scale population of researchers/practitioners interested on [the](#) improve the features of the Issue Tracking Systems. Based on your area of interest, we kindly invite you to take part in the following survey:

{{url do formulario}}

You [were](#) chosen because your relevant participation/contribution in {{nome da fonte de amostragem}}- {{url da fonte de amostragem}}. Your opinion is essential to strength our findings. Please, help us accordingly [to](#) your possibilities by [answering](#) this survey until {{data limite}}. As soon as we conclude [the](#) data analysis, we will share the results with all participants and the software engineering community. If you have already fulfilled this questionnaire, please ignore this email.

Thanks in advance,
Vagner Clementino

4.4 Resultados

Neste seção apresentamos os resultados obtidos da aplicação do questionário. Começamos com a análise do perfil dos respondentes. Em seguida, avaliamos o nível de satisfação que os participantes possuem com as ferramentas que eles utilizam. Posteriormente verificamos a adoção das metodologias propostas pelos agilistas no processo de desenvolvimento e em especial da manutenção de software.

4.4.1 Perfil dos Participantes

Antes de apresentarmos os resultado sobre as ferramentas, avaliamos o perfil dos respondentes. Como pode ser observado na Figura 4.4 a função mais frequente é a de desenvolvedor. Todavia, grande parte dos respondentes estão diretamente vinculados ao desenvolvimento e manutenção de software, tanto que mais de 80% da amostrada é formada por desenvolvedores, engenheiros de software, gerentes e arquitetos. Neste sentido, pelo menos no que tange à função desempenhada, obtivemos um conjunto significativo de participantes.

A distribuição geográfica dos participantes pode ser visualizada na Figura 4.5. Há

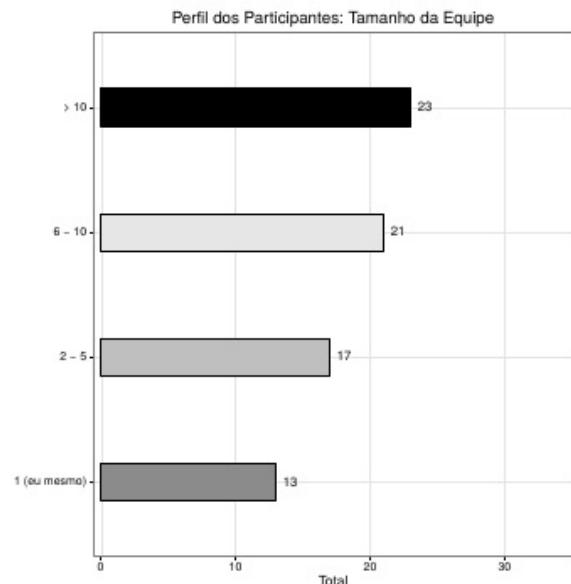


Figura 4.7: Tamanho da Equipe

que utiliza para outro projeto. A probabilidade de recomendação é exibida na Figura 4.11. De maneira similar ao nível de satisfação grande parte dos participantes tendem a recomendar a FGRM. Com base neste resultado, podemos deduzir que os profissionais estão realmente satisfeitos com as funcionalidades da ferramenta que utiliza ao ponto de recomendá-la.

4.4.3 Avaliação das Funcionalidades Existentes

Nesta seção apresentamos a opinião dos profissionais sobre as funcionalidades oferecidas atualmente pelas FGRM. O conjunto de funcionalidades apresentado ao participante é o resultado do estudo descrito na Seção ???. Este ponto de vista pode ser visualizado na Tabela 4.2. É possível verificar que os profissionais avaliaram como importantes funções tais como *Supporte ao Unicode*, *Múltiplos Projetos*, *Integração com Sistemas de Controle de Versão (VCS Integration)* como funções importantes em sua atividades diárias.

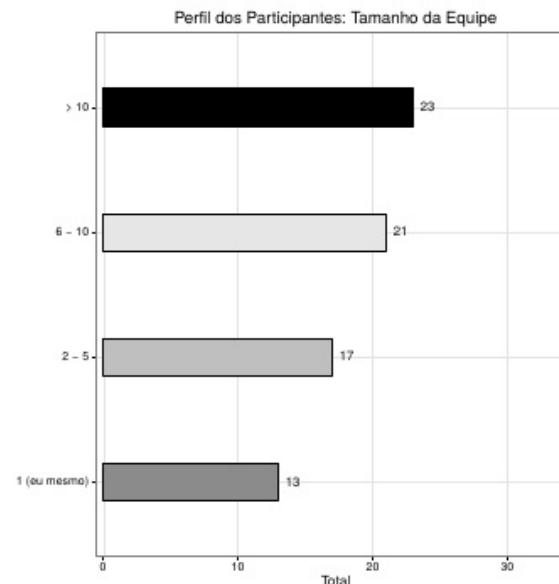


Figura 4.7: Tamanho da Equipe

que utiliza para outro projeto. A probabilidade de recomendação é exibida na Figura 4.11. De maneira similar ao nível de satisfação grande parte dos participantes tendem a recomendar a FGRM. Com base neste resultado, podemos deduzir que os profissionais estão realmente satisfeitos com as funcionalidades da ferramenta que utiliza ao ponto de recomendá-la.

4.4.3 Avaliação das Funcionalidades Existentes

Nesta seção apresentamos a opinião dos profissionais sobre as funcionalidades oferecidas atualmente pelas FGRM. O conjunto de funcionalidades apresentado ao participante é o resultado do estudo descrito na Seção ???. Este ponto de vista pode ser visualizado na Tabela 4.2. É possível verificar que os profissionais avaliaram como importantes funções tais como *Supporte ao Unicode*, *Múltiplos Projetos*, *Integração com Sistemas de Controle de Versão (VCS Integration)* como funções importantes em sua atividades diárias.

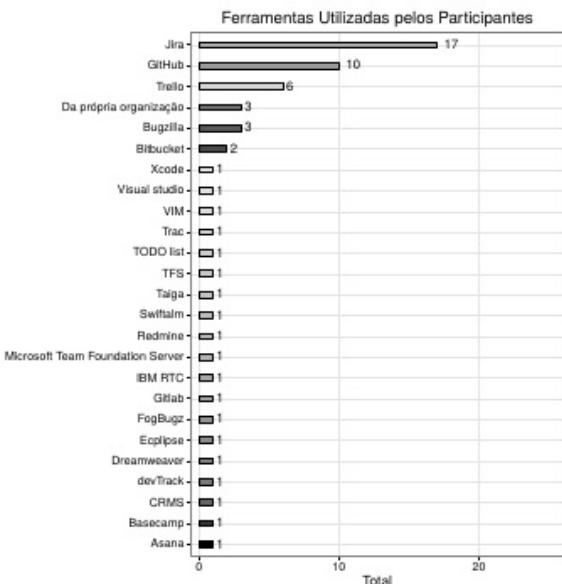


Figura 4.9: Ferramentas utilizadas pelos participantes

Por outro lado, apresentamos aos profissionais funcionalidades que poderiam ser integradas à FGRM que ele utiliza. A opinião pode ser visualizada na Figura 4.13. **Conforme** pode ser observado melhorias na **busca de** busca de RMs, coleta de informações para solucionar na resolução da RM e suporte ao registro de RMs foram avaliadas como funcionalidades que podem melhorar as atividades do desenvolvedor.

Algumas das melhorias propostas na literatura se mostraram interessantes pelos profissionais. A Figura 4.12 apresenta as funcionalidades que os participantes sentem falta. Funções tais como identificação automática de RMs duplicadas, atribuição automática de RM e Análise de Impacto foram as mais frequentes.

4.4.4 Práticas Ágeis na Manutenção de Software

Nesta etapa do levantamento com profissionais, estamos interessados em analisar como as práticas propostas pelos agilistas estão sendo utilizadas no desenvolvimento e em especial na manutenção de software. A Figura 4.14 exibe, segundo os participantes,

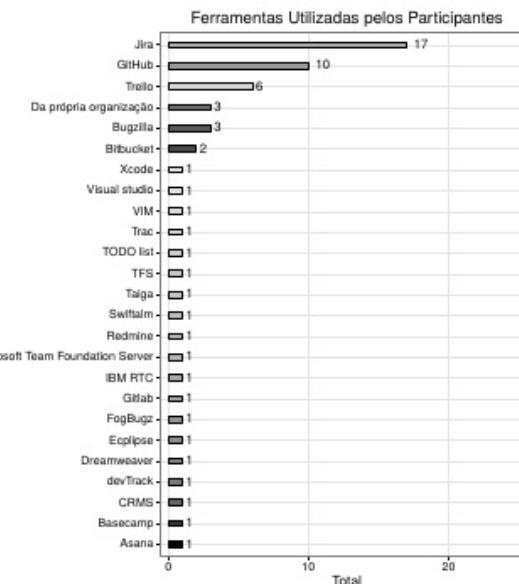


Figura 4.9: Ferramentas utilizadas pelos participantes

Por outro lado, apresentamos aos profissionais funcionalidades que poderiam ser integradas à FGRM que ele utiliza. A opinião pode ser visualizada na Figura 4.13. **Conforme** pode ser observado melhorias na busca de RMs, coleta de informações para solucionar na resolução da RM e suporte ao registro de RMs foram avaliadas como funcionalidades que podem melhorar as atividades do desenvolvedor.

Algumas das melhorias propostas na literatura se mostraram interessantes pelos profissionais. A Figura 4.12 apresenta as funcionalidades que os participantes sentem falta. Funções tais como identificação automática de RMs duplicadas, atribuição automática de RM e Análise de Impacto foram as mais frequentes.

4.4.4 Práticas Ágeis na Manutenção de Software

Nesta etapa do levantamento com profissionais, estamos interessados em analisar como as práticas propostas pelos agilistas estão sendo utilizadas no desenvolvimento e em especial na manutenção de software. A Figura 4.14 exibe, segundo os participantes,

4.5. DISCUSSÃO

89

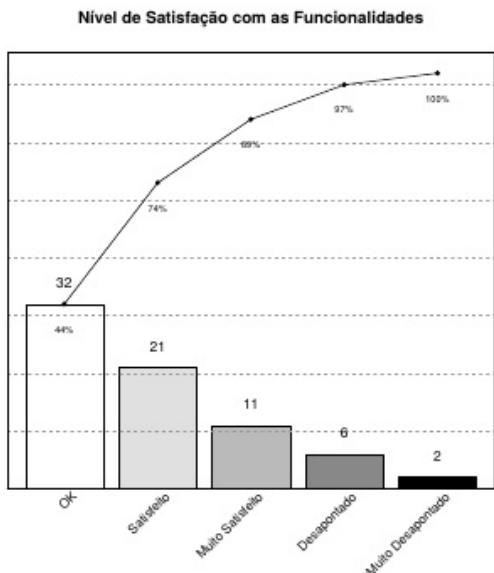


Figura 4.10: Nível de satisfação com as Ferramentas

as práticas das metodologias ágeis que estavam sendo utilizadas. Dentre elas, as mais adotadas foram Integração Contínua, Padrões de Programação e Refatoração.

A fim de avaliar como as FGRMs podem ajudar aos times de manutenção de software na adoção das práticas propostas agilistas, apresentamos aos participantes do levantamento uma lista de possíveis funcionalidades com este viés. A Figura 4.15 apresenta a opinião dos profissionais sobre as funcionalidades mais relevantes. Segundo eles, a priorização automática de RMs urgente e não esperadas, ajuda do desenvolvedor em sua reunião diária (daily) e o suporte a tarefas compartilhadas foram as respostas mais frequentes.

4.5 Discussão

Nível de Satisfação com a Ferramenta Utilizada. Em geral, o nível de satisfação com as funcionalidades oferecidas pelas FGRMs é alto. Esta medida foi observada no

4.5. DISCUSSÃO

89

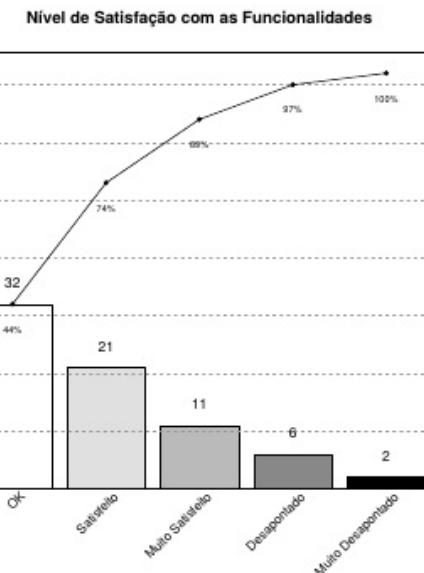


Figura 4.10: Nível de satisfação com as Ferramentas

as práticas das metodologias ágeis que estavam sendo utilizadas. Dentre elas, as mais adotadas foram Integração Contínua, Padrões de Programação e Refatoração.

A fim de avaliar como as FGRMs podem ajudar aos times de manutenção de software na adoção das práticas propostas agilistas, apresentamos aos participantes do levantamento uma lista de possíveis funcionalidades com este viés. A Figura 4.15 apresenta a opinião dos profissionais sobre as funcionalidades mais relevantes. Segundo eles, a priorização automática de RMs urgente e não esperadas, ajuda do desenvolvedor em sua reunião diária (daily) e o suporte a tarefas compartilhadas foram as respostas mais frequentes.

4.5 Discussão

Nível de Satisfação com a Ferramenta Utilizada: Em geral, o nível de satisfação com as funcionalidades oferecidas pelas FGRMs é alto. Esta medida foi observada no

Probabilidade de Recomendação da FGRM Utilizada

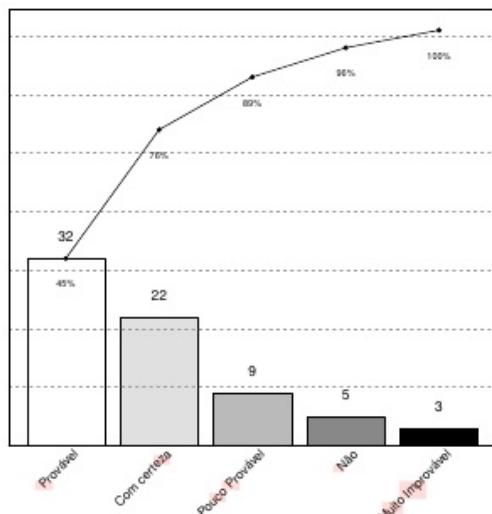


Figura 4.11: Probabilidade de Recomendação da Ferramenta Utilizada

Figura 4.10 no qual verificamos que cerca de 90% dos participantes estão de alguma forma satisfeitos com a ferramenta que utiliza. Este mesmo sentimento pode ser observado pela relativamente alta probabilidade de recomendação da FGRM utilizada para um novo projeto. Naquela medida verificamos que o mesmo percentual de participantes pretendem recomendar o software que utiliza.

Funcionalidades Faltantes.

Supporte às Práticas dos Agilistas

4.6 Ameças à Validade

A principal ameaça à validade deste trabalho está no numero de respondentes da pesquisa. Apesar de ter sido realizada uma seleção metodológica de uma amostra

Probabilidade de Recomendação da FGRM Utilizada

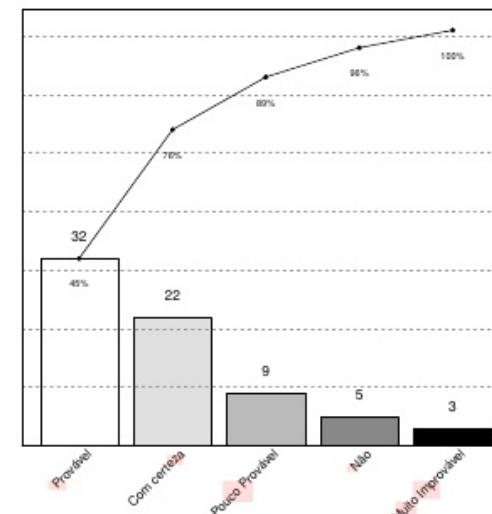


Figura 4.11: Probabilidade de Recomendação da Ferramenta Utilizada

Figura 4.10 no qual verificamos que cerca de 90% dos participantes estão de alguma forma satisfeitos com a ferramenta que utiliza. Este mesmo sentimento pode ser observado pela relativamente alta probabilidade de recomendação da FGRM utilizada para um novo projeto. Naquela medida verificamos que o mesmo percentual de participantes pretendem recomendar o software que utiliza.

Funcionalidades Faltantes: Apesar dos profissionais estarem satisfeitos com as funcionalidades oferecidas pela FGRM que utiliza, quando lhe foi apresentado um conjunto de novas funções grande parte dos participantes aprova a inclusão de algumas delas. Por exemplo, certa de um terço dos participantes disseram sentir falta de um processo de identificação automática de RMs duplicadas. Este resultado também foi encontrado no trabalho de Bettenburg e outros [Bettenburg et al., 2008b] que ao conduzir um levantamento com questionário onde o problema da duplicação de RMs foi descrito com um dos problemas que pode atrapalhar o processo de solução da requisição.

Um outro ponto interessante a destacar é que as funcionalidades mais sentem

4.6. AMEÇAS À VALIDADE

91

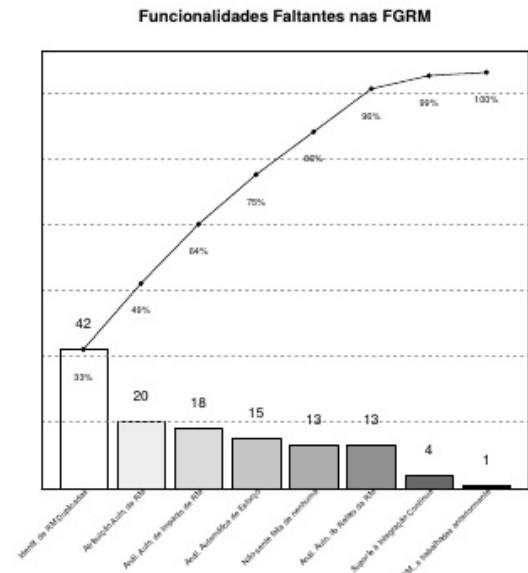


Figura 4.12: Funcionalidades que o participantes sentem falta.

Item	Overall Rank	Rank Distribution	Score	No. of Rankings
The CRMS should provide a powerful, yet simple and easy-to-use feature to search bug reports.	1	■ ■ ■ ■ ■	197	48
The CRMS should provide support for users to collect and prepare information that developers need.	2	■ ■ ■ ■ ■	190	44
The CRMS should give cues to inexperienced reporters that information they should provide and how they can collect it.	3	■ ■ ■ ■ ■	164	42
The CRMS should integrate reputation into user profiles to mark experienced reporters.	4	■ ■ ■ ■ ■	135	45
The CRMS should reward reporters, when they do a good report.	5	■ ■ ■ ■ ■	130	44
The CRMS should provide support to translate bug reports filed in foreign languages.	6	■ ■ ■ ■ ■	128	43

■ ■ ■ ■ ■
Lowest Rank Highest Rank

Figura 4.13: Novas funcionalidades para as FGRMs.

4.5. DISCUSSÃO

91

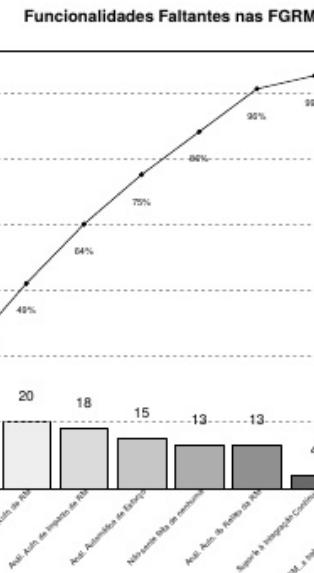


Figura 4.12: Funcionalidades que o participantes sentem falta.

Item	Overall Rank	Rank Distribution	Score	No. of Rankings
The CRMS should provide a powerful, yet simple and easy-to-use feature to search bug reports.	1	■ ■ ■ ■ ■	197	48
The CRMS should provide support for users to collect and prepare information that developers need.	2	■ ■ ■ ■ ■	190	44
The CRMS should give cues to inexperienced reporters that information they should provide and how they can collect it.	3	■ ■ ■ ■ ■	164	42
The CRMS should integrate reputation into user profiles to mark experienced reporters.	4	■ ■ ■ ■ ■	135	45
The CRMS should reward reporters, when they do a good report.	5	■ ■ ■ ■ ■	130	44
The CRMS should provide support to translate bug reports filed in foreign languages.	6	■ ■ ■ ■ ■	128	43

■ ■ ■ ■ ■
Lowest Rank Highest Rank

Figura 4.13: Novas funcionalidades para as FGRMs.

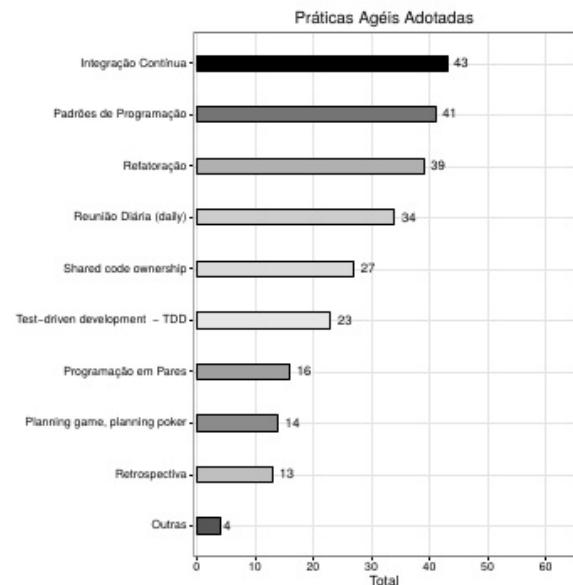


Figura 4.14: Metodologias propostas pelos agilistas que são adotadas pelos participantes.

representativa da população o número de participantes limita a extração do resultado obtido.

Non-probability samples are limited with regard to generalization. Because they do not truly represent a population, we cannot make valid inferences about the larger group from which they are drawn. Validity can be increased by approximating random selection as much as possible, and making every attempt to avoid introducing bias into sample selection.

The most obvious criticism about convenience sampling is sampling bias and that the sample is not representative of the entire population. This may be the biggest disadvantage when using a convenience sample because it leads to more problems and criticisms.

Systematic bias stems from sampling bias. This refers to a constant difference between the results from the sample and the theoretical results from the entire population. It is not rare that the results from a study that uses a convenience sample

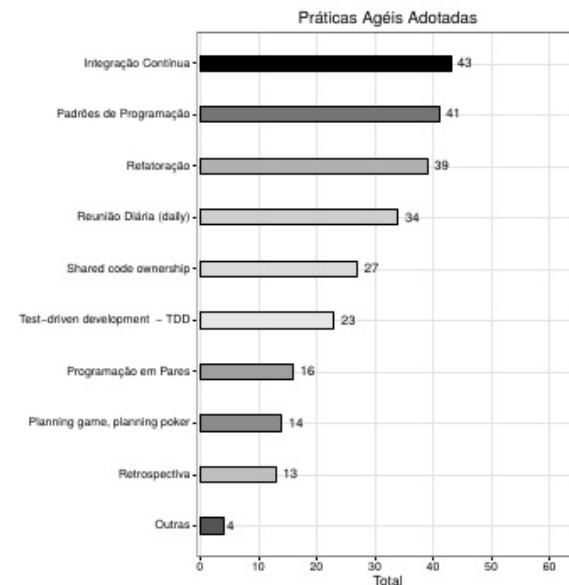


Figura 4.14: Metodologias propostas pelos agilistas que são adotadas pelos participantes.

falta, também representam a maior quantidade de estudos na literatura. Posto de outra forma, a automatização da detecção de duplicadas e atribuição de RMs são os mais demandados e representam o maior número de trabalhos na literatura. Este resultado pode sugerir a necessidade de divulgação do que está sendo proposto na literatura tendo em vista que os profissionais se mostraram interessados nestes tipos de funcionalidades.

Suporte às Práticas dos Agilistas: Apesar de ser pouco discutido na literatura, as FGRMs podem oferecer suporte às práticas propostas pelos agilistas. Os participantes se mostrarem interessados em funções tais como a priorização automática de RMs urgente e não esperadas, ajuda do desenvolvedor em sua reunião diária (daily) e o suporte a tarefas compartilhadas. Com a crescente adoção das práticas dos agilistas por times de desenvolvimento e manutenção de software seria importante que este tipo de ferramenta incorporasse em suas funcionalidades tal tendência. Segundo o nosso

4.6. AMEÇAS À VALIDADE

93

Item	Overall Rank	Rank Distribution	Score	No. of Rankings
Automatic prioritization of urgent or unexpected Change Request	1	██████	164	33
Help developer in preparation for the daily meeting.	2	██████	149	38
Support to sharing tasks	3	██████	148	34
The automated suggestion of Change Request that will be part of the iteration.	4	██████	135	32
Facilitate shared ownership of code through the ability of developers to test each other's code, that is, avoid code's ownership.	5	██████	133	34
Support for the design of use cases for purposes of communication among developers.	6	██████	123	32
Support a structure of multiclient relationship	7	██████	98	31

Legend:
 █ █ █ █ █ Lowest Rank
 █ █ █ █ █ Highest Rank

Figura 4.15: Classificação das funcionalidades que possam dar suporte ao uso das metodologias dos agilistas.

differ significantly with the results from the entire population. A consequence of having systematic bias is obtaining skewed results.

Another significant criticism about using a convenience sample is the limitation in generalization and inference making about the entire population. Since the sample is not representative of the population, the results of the study cannot speak for the entire population. This results to a low external validity of the study.

When using convenience sampling, it is necessary to describe how your sample would differ from an ideal sample that was randomly selected. It is also necessary to describe the individuals who might be left out during the selection process or the individuals who are overrepresented in the sample.

In connection to this, it is better if you can describe the possible effects of the people who were left out or the subjects that are overrepresented to your results. This will allow the readers of your research to get a good grasp of the sample that you were testing. It will also enable them to estimate the possible difference between your results and the results from the entire population.

Vale ressaltar que todas as opiniões coletadas devem sempre ser levadas em conta a ferramenta que o profissional utilizava quando da aplicação do questionário. Caso este mesmo estudo fosse realizado com outras versões do mesmo sistema os resultados poderiam ser diferentes. Neste sentido, a generalização dos resultados passa por esta característica do estudo.

4.6. AMEÇAS À VALIDADE

93

Item	Overall Rank	Rank Distribution	Score	No. of Rankings
Automatic prioritization of urgent or unexpected Change Request	1	██████	164	33
Help developer in preparation for the daily meeting.	2	██████	149	38
Support to sharing tasks	3	██████	148	34
The automated suggestion of Change Request that will be part of the iteration.	4	██████	135	32
Facilitate shared ownership of code through the ability of developers to test each other's code, that is, avoid code's ownership.	5	██████	133	34
Support for the design of use cases for purposes of communication among developers.	6	██████	123	32
Support a structure of multiclient relationship	7	██████	98	31

Legend:
 █ █ █ █ █ Lowest Rank
 █ █ █ █ █ Highest Rank

Figura 4.15: Classificação das funcionalidades que possam dar suporte ao uso das metodologias dos agilistas.

atendimento e discutido na Seção 2.3 as FGRMs estão longe de atender as demandas dos agilistas.

4.6 Ameças à Validade

A principal ameaça à validade deste trabalho está no numero de respondentes da pesquisa. Apesar de ter sido realizada uma seleção metodológica de uma amostra representativa da população o número de participantes limita a extração dos resultados obtidos. O fato de ter sido utilizada uma amostragem de conveniência, as generalizações são limitada já que amostra não verdadeiramente representa a população. Por outro lado, utilizamos o arcabouço proposto por de Mello [de Mello et al., 2014] visando minimizar a introdução de viés na amostra selecionada.

Ainda avaliando o processo de seleção das amostras utilizadas, não temos garantia que as regras para seleção de participantes resultaram no conjunto mais representativo da população. Vale ressaltar que todas as opiniões coletadas devem sempre ser levadas em conta a ferramenta que o profissional utilizava quando da aplicação do questionário. Caso este mesmo estudo fosse realizado com outras versões do mesmo sistema os resultados poderiam ser diferentes. Neste sentido, a generalização dos resultados passa por esta característica do estudo.

94 CAPÍTULO 4. LEVANTAMENTO POR QUESTIONÁRIO COM PROFISSIONAIS

característica do estudo.

4.7 Resumo do Capítulo

94 CAPÍTULO 4. LEVANTAMENTO POR QUESTIONÁRIO COM PROFISSIONAIS

4.7 Resumo do Capítulo

REFERÊNCIAS BIBLIOGRÁFICAS

113

- [Bettenburg et al., 2008a] Bettenburg, N.; Just, S.; Schröter, A.; Weiss, C.; Premraj, R. & Zimmermann, T. (2008a). What makes a good bug report? Em *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pp. 308–318. ACM.
- [Bettenburg et al., 2008b] Bettenburg, N.; Just, S.; Schröter, A.; Weiss, C.; Premraj, R. & Zimmermann, T. (2008b). What makes a good bug report? Em *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pp. 308–318. ACM.
- [Bettenburg et al., 2008c] Bettenburg, N.; Premraj, R.; Zimmermann, T. & Kim, S. (2008c). Duplicate bug reports considered harmful... really? Em *Software maintenance, 2008. ICSM 2008. IEEE international conference on*, pp. 337–345. IEEE.
- [Bhattacharya & Neamtiu, 2011] Bhattacharya, P. & Neamtiu, I. (2011). Bug-fix time prediction models: can we do better? Em *Proceedings of the 8th Working Conference on Mining Software Repositories*, pp. 207–210. ACM.
- [Bird et al., 2009] Bird, C.; Rigby, P. C.; Barr, E. T.; Hamilton, D. J.; German, D. M. & Devanbu, P. (2009). The promises and perils of mining git. *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR 2009*, pp. 1–10. ISSN 15737616.
- [Breu et al., 2010a] Breu, S.; Premraj, R.; Sillito, J. & Zimmermann, T. (2010a). Information needs in bug reports: Improving cooperation between developers and users. Em *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW '10*, pp. 301–310, New York, NY, USA. ACM.
- [Breu et al., 2010b] Breu, S.; Premraj, R.; Sillito, J. & Zimmermann, T. (2010b). Information needs in bug reports: improving cooperation between developers and users. Em *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pp. 301–310. ACM.
- [Cavalcanti et al., 2014] Cavalcanti, Y. C.; Mota Silveira Neto, P. A.; Machado, I. d. C.; Vale, T. F.; Almeida, E. S. & Meira, S. R. d. L. (2014). Challenges and opportunities for software change request repositories: a systematic mapping study. *Journal of Software: Evolution and Process*, 26(7):620–653.
- [Cavalcanti et al., 2013] Cavalcanti, Y. C.; Neto, P. A. d. M. S.; Lucrédio, D.; Vale, T.; de Almeida, E. S. & de Lemos Meira, S. R. (2013). The bug report duplication problem: an exploratory study. *Software Quality Journal*, 21(1):39–66.

REFERÊNCIAS BIBLIOGRÁFICAS

113

- [Bettenburg et al., 2008a] Bettenburg, N.; Just, S.; Schröter, A.; Weiss, C.; Premraj, R. & Zimmermann, T. (2008a). What makes a good bug report? Em *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pp. 308–318. ACM.
- [Bettenburg et al., 2008b] Bettenburg, N.; Just, S.; Schröter, A.; Weiss, C.; Premraj, R. & Zimmermann, T. (2008b). What makes a good bug report? Em *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pp. 308–318. ACM.
- [Bettenburg et al., 2008c] Bettenburg, N.; Premraj, R.; Zimmermann, T. & Kim, S. (2008c). Duplicate bug reports considered harmful... really? Em *Software maintenance, 2008. ICSM 2008. IEEE international conference on*, pp. 337–345. IEEE.
- [Bhattacharya & Neamtiu, 2011] Bhattacharya, P. & Neamtiu, I. (2011). Bug-fix time prediction models: can we do better? Em *Proceedings of the 8th Working Conference on Mining Software Repositories*, pp. 207–210. ACM.
- [Bird et al., 2009] Bird, C.; Rigby, P. C.; Barr, E. T.; Hamilton, D. J.; German, D. M. & Devanbu, P. (2009). The promises and perils of mining git. *Proceedings of the 2009 6th IEEE International Working Conference on Mining Software Repositories, MSR 2009*, pp. 1–10. ISSN 15737616.
- [Boxill et al., 1997] Boxill, I.; Chambers, C. M. & Wint, E. (1997). *Introduction to social research: With applications to the Caribbean*. University of The West Indies Press.
- [Breu et al., 2010a] Breu, S.; Premraj, R.; Sillito, J. & Zimmermann, T. (2010a). Information needs in bug reports: Improving cooperation between developers and users. Em *Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW '10*, pp. 301–310, New York, NY, USA. ACM.
- [Breu et al., 2010b] Breu, S.; Premraj, R.; Sillito, J. & Zimmermann, T. (2010b). Information needs in bug reports: improving cooperation between developers and users. Em *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pp. 301–310. ACM.
- [Cavalcanti et al., 2014] Cavalcanti, Y. C.; Mota Silveira Neto, P. A.; Machado, I. d. C.; Vale, T. F.; Almeida, E. S. & Meira, S. R. d. L. (2014). Challenges and opportunities for software change request repositories: a systematic mapping study. *Journal of Software: Evolution and Process*, 26(7):620–653.

- [Cerulo & Canfora, 2004] Cerulo, L. & Canfora, G. (2004). A taxonomy of information retrieval models and tools. *CIT. Journal of computing and information technology*, 12(3):175–194.
- [Chawla & Singh, 2015] Chawla, I. & Singh, S. K. (2015). An automated approach for bug categorization using fuzzy logic. Em *Proceedings of the 8th India Software Engineering Conference*, pp. 90–99. ACM.
- [Choudhari & Suman, 2014] Choudhari, J. & Suman, U. (2014). Extended iterative maintenance life cycle using extreme programming. *SIGSOFT Softw. Eng. Notes*, 39(1):1–12. ISSN 0163-5948.
- [Corley et al., 2011] Corley, C. S.; Kraft, N. A.; Etzkorn, L. H. & Lukins, S. K. (2011). Recovering traceability links between source code and fixed bugs via patch analysis. Em *Proceedings of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 31–37. ACM.
- [Correa et al., 2013] Correa, D.; Lal, S.; Saini, A. & Sureka, A. (2013). Samekana: A browser extension for including relevant web links in issue tracking system discussion forum. Em *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, volume 1, pp. 25–33. IEEE.
- [Creswell & Clark, 2007] Creswell, J. W. & Clark, V. L. P. (2007). Designing and conducting mixed methods research.
- [Dal Sasse & Lanza, 2013] Dal Sasse, T. & Lanza, M. (2013). A closer look at bugs. Em *Software Visualization (VISSOFT), 2013 First IEEE Working Conference on*, pp. 1–4. IEEE.
- [Dal Sasso & Lanza, 2014] Dal Sasso, T. & Lanza, M. (2014). In* bug: Visual analytics of bug repositories. Em *Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week-IEEE Conference on*, pp. 415–419. IEEE.
- [de Mello et al., 2014] de Mello, R. M.; da Silva, P. C.; Runeson, P. & Travassos, G. H. (2014). Towards a framework to support large scale sampling in software engineering surveys. Em *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 48. ACM.
- [de Mello et al., 2015] de Mello, R. M.; Da Silva, P. C. & Travassos, G. H. (2015). Investigating probabilistic sampling approaches for large-scale surveys in software engineering. *Journal of Software Engineering Research and Development*, 3(1):8.

- [Cavalcanti et al., 2013] Cavalcanti, Y. C.; Neto, P. A. d. M. S.; Lucrédio, D.; Vale, T.; de Almeida, E. S. & de Lemos Meira, S. R. (2013). The bug report duplication problem: an exploratory study. *Software Quality Journal*, 21(1):39–66.
- [Cerulo & Canfora, 2004] Cerulo, L. & Canfora, G. (2004). A taxonomy of information retrieval models and tools. *CIT. Journal of computing and information technology*, 12(3):175–194.
- [Chawla & Singh, 2015] Chawla, I. & Singh, S. K. (2015). An automated approach for bug categorization using fuzzy logic. Em *Proceedings of the 8th India Software Engineering Conference*, pp. 90–99. ACM.
- [Choudhari & Suman, 2014] Choudhari, J. & Suman, U. (2014). Extended iterative maintenance life cycle using extreme programming. *SIGSOFT Softw. Eng. Notes*, 39(1):1–12. ISSN 0163-5948.
- [Corley et al., 2011] Corley, C. S.; Kraft, N. A.; Etzkorn, L. H. & Lukins, S. K. (2011). Recovering traceability links between source code and fixed bugs via patch analysis. Em *Proceedings of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering*, pp. 31–37. ACM.
- [Correa et al., 2013] Correa, D.; Lal, S.; Saini, A. & Sureka, A. (2013). Samekana: A browser extension for including relevant web links in issue tracking system discussion forum. Em *2013 20th Asia-Pacific Software Engineering Conference (APSEC)*, volume 1, pp. 25–33. IEEE.
- [Creswell & Clark, 2007] Creswell, J. W. & Clark, V. L. P. (2007). Designing and conducting mixed methods research.
- [Dal Sasse & Lanza, 2013] Dal Sasse, T. & Lanza, M. (2013). A closer look at bugs. Em *Software Visualization (VISSOFT), 2013 First IEEE Working Conference on*, pp. 1–4. IEEE.
- [Dal Sasso & Lanza, 2014] Dal Sasso, T. & Lanza, M. (2014). In* bug: Visual analytics of bug repositories. Em *Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE), 2014 Software Evolution Week-IEEE Conference on*, pp. 415–419. IEEE.
- [de Mello et al., 2014] de Mello, R. M.; da Silva, P. C.; Runeson, P. & Travassos, G. H. (2014). Towards a framework to support large scale sampling in software engineering surveys. Em *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 48. ACM.

REFERÊNCIAS BIBLIOGRÁFICAS

115

- [de Mello & Travassos, 2013] de Mello, R. M. & Travassos, G. H. (2013). Would sociable software engineers observe better? Em *Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on*, pp. 279–282. IEEE.
- [Devulapally, 2015] Devulapally, G. K. (2015). Agile in the context of Software Maintainability.
- [Di Lucca et al., 2002] Di Lucca, G. A.; Di Penta, M. & Gradara, S. (2002). An approach to classify software maintenance requests. Em *Software Maintenance, 2002. Proceedings. International Conference on*, pp. 93–102. IEEE.
- [Dybå et al., 2007] Dybå, T.; Dingsøyr, T. & Hanssen, G. K. (2007). Applying systematic reviews to diverse study types: An experience report. Em *ESEM*, volume 7, pp. 225–234.
- [Dybå et al., 2006] Dybå, T.; Kampenes, V. B. & Sjøberg, D. I. (2006). A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, 48(8):745–755.
- [Engelbertink & Vogt, 2010] Engelbertink, F. P. & Vogt, H. H. (2010). How to save on software maintenance costs. *Omnex White Paper. Accessed*.
- [Etikan et al., 2016] Etikan, I.; Musa, S. A. & Alkassim, R. S. (2016). Comparison of convenience sampling and purposive sampling. *American Journal of Theoretical and Applied Statistics*, 5(1):1–4.
- [Fan & Yan, 2010] Fan, W. & Yan, Z. (2010). Factors affecting response rates of the web survey: A systematic review. *Computers in human behavior*, 26(2):132–139.
- [Fox et al., 2013] Fox, A.; Patterson, D. A. & Joseph, S. (2013). *Engineering software as a service: an agile approach using cloud computing*. Strawberry Canyon LLC.
- [Gegick et al., 2010] Gegick, M.; Rotella, P. & Xie, T. (2010). Identifying security bug reports via text mining: An industrial case study. Em *2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010)*, pp. 11–20. IEEE.
- [Heeager & Rose, 2015] Heeager, L. T. & Rose, J. (2015). Optimising agile development practices for the maintenance operation: nine heuristics. *Empirical Software Engineering*, 20(6):1762–1784. ISSN 15737616.
- [Herrin, 1985] Herrin, W. R. (1985). Software maintenance costs: A quantitative evaluation. Em *Proceedings of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, SIGCSE '85*, pp. 233–237, New York, NY, USA. ACM.

REFERÊNCIAS BIBLIOGRÁFICAS

115

- [de Mello et al., 2015] de Mello, R. M.; Da Silva, P. C. & Travassos, G. H. (2015). Investigating probabilistic sampling approaches for large-scale surveys in software engineering. *Journal of Software Engineering Research and Development*, 3(1):8.
- [de Mello & Travassos, 2013] de Mello, R. M. & Travassos, G. H. (2013). Would sociable software engineers observe better? Em *Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on*, pp. 279–282. IEEE.
- [Devulapally, 2015] Devulapally, G. K. (2015). Agile in the context of Software Maintainability.
- [Di Lucca et al., 2002] Di Lucca, G. A.; Di Penta, M. & Gradara, S. (2002). An approach to classify software maintenance requests. Em *Software Maintenance, 2002. Proceedings. International Conference on*, pp. 93–102. IEEE.
- [Dybå et al., 2007] Dybå, T.; Dingsøyr, T. & Hanssen, G. K. (2007). Applying systematic reviews to diverse study types: An experience report. Em *ESEM*, volume 7, pp. 225–234.
- [Dybå et al., 2006] Dybå, T.; Kampenes, V. B. & Sjøberg, D. I. (2006). A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, 48(8):745–755.
- [Engelbertink & Vogt, 2010] Engelbertink, F. P. & Vogt, H. H. (2010). How to save on software maintenance costs. *Omnex White Paper. Accessed*.
- [Fan & Yan, 2010] Fan, W. & Yan, Z. (2010). Factors affecting response rates of the web survey: A systematic review. *Computers in human behavior*, 26(2):132–139.
- [Fox et al., 2013] Fox, A.; Patterson, D. A. & Joseph, S. (2013). *Engineering software as a service: an agile approach using cloud computing*. Strawberry Canyon LLC.
- [Gegick et al., 2010] Gegick, M.; Rotella, P. & Xie, T. (2010). Identifying security bug reports via text mining: An industrial case study. Em *2010 7th IEEE Working Conference on Mining Software Repositories (MSR 2010)*, pp. 11–20. IEEE.
- [Heeager & Rose, 2015] Heeager, L. T. & Rose, J. (2015). Optimising agile development practices for the maintenance operation: nine heuristics. *Empirical Software Engineering*, 20(6):1762–1784. ISSN 15737616.
- [Herrin, 1985] Herrin, W. R. (1985). Software maintenance costs: A quantitative evaluation. Em *Proceedings of the Sixteenth SIGCSE Technical Symposium on Computer Science Education, SIGCSE '85*, pp. 233–237, New York, NY, USA. ACM.

REFERÊNCIAS BIBLIOGRÁFICAS

119

- [Maiden & Rugg, 1996] Maiden, N. A. & Rugg, G. (1996). Acre: selecting methods for requirements acquisition. *Software Engineering Journal*, 11(3):183–192.
- [Malheiros et al., 2012] Malheiros, Y.; Moraes, A.; Trindade, C. & Meira, S. (2012). A source code recommender system to support newcomers. Em *2012 IEEE 36th Annual Computer Software and Applications Conference*, pp. 19–24. IEEE.
- [Mani et al., 2012] Mani, S.; Catherine, R.; Sinha, V. S. & Dubey, A. (2012). Ausum: approach for unsupervised bug report summarization. Em *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, p. 11. ACM.
- [McGee & Greer, 2009] McGee, S. & Greer, D. (2009). A software requirements change source taxonomy. Em *Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on*, pp. 51–58. IEEE.
- [Moran, 2015] Moran, K. (2015). Enhancing android application bug reporting. Em *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 1045–1047. ACM.
- [Moran et al., 2015] Moran, K.; Linares-Vásquez, M.; Bernal-Cárdenas, C. & Poshyvanyk, D. (2015). Auto-completing bug reports for android applications. Em *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 673–686. ACM.
- [Naguib et al., 2013] Naguib, H.; Narayan, N.; Brügge, B. & Helal, D. (2013). Bug report assignee recommendation using activity profiles. Em *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, pp. 22–30. IEEE.
- [Nagwani et al., 2013] Nagwani, N.; Verma, S. & Mehta, K. K. (2013). Generating taxonomic terms for software bug classification by utilizing topic models based on latent dirichlet allocation. Em *ICT and Knowledge Engineering (ICT&KE), 2013 11th International Conference on*, pp. 1–5. IEEE.
- [Nagwani & Verma, 2010a] Nagwani, N. K. & Verma, S. (2010a). Predictive data mining model for software bug estimation using average weighted similarity. Em *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, pp. 373–378. IEEE.
- [Nagwani & Verma, 2010b] Nagwani, N. K. & Verma, S. (2010b). Predictive data mining model for software bug estimation using average weighted similarity. Em *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, pp. 373–378. IEEE.

REFERÊNCIAS BIBLIOGRÁFICAS

119

- [Maiden & Rugg, 1996] Maiden, N. A. & Rugg, G. (1996). Acre: selecting methods for requirements acquisition. *Software Engineering Journal*, 11(3):183–192.
- [Malheiros et al., 2012] Malheiros, Y.; Moraes, A.; Trindade, C. & Meira, S. (2012). A source code recommender system to support newcomers. Em *2012 IEEE 36th Annual Computer Software and Applications Conference*, pp. 19–24. IEEE.
- [Mani et al., 2012] Mani, S.; Catherine, R.; Sinha, V. S. & Dubey, A. (2012). Ausum: approach for unsupervised bug report summarization. Em *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, p. 11. ACM.
- [Marshall, 1996] Marshall, M. N. (1996). Sampling for qualitative research. *Family practice*, 13(6):522–526.
- [McGee & Greer, 2009] McGee, S. & Greer, D. (2009). A software requirements change source taxonomy. Em *Software Engineering Advances, 2009. ICSEA'09. Fourth International Conference on*, pp. 51–58. IEEE.
- [Moran, 2015] Moran, K. (2015). Enhancing android application bug reporting. Em *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 1045–1047. ACM.
- [Moran et al., 2015] Moran, K.; Linares-Vásquez, M.; Bernal-Cárdenas, C. & Poshyvanyk, D. (2015). Auto-completing bug reports for android applications. Em *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 673–686. ACM.
- [Naguib et al., 2013] Naguib, H.; Narayan, N.; Brügge, B. & Helal, D. (2013). Bug report assignee recommendation using activity profiles. Em *Mining Software Repositories (MSR), 2013 10th IEEE Working Conference on*, pp. 22–30. IEEE.
- [Nagwani et al., 2013] Nagwani, N.; Verma, S. & Mehta, K. K. (2013). Generating taxonomic terms for software bug classification by utilizing topic models based on latent dirichlet allocation. Em *ICT and Knowledge Engineering (ICT&KE), 2013 11th International Conference on*, pp. 1–5. IEEE.
- [Nagwani & Verma, 2010a] Nagwani, N. K. & Verma, S. (2010a). Predictive data mining model for software bug estimation using average weighted similarity. Em *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, pp. 373–378. IEEE.

- [Nagwani & Verma, 2012] Nagwani, N. K. & Verma, S. (2012). Predicting expert developers for newly reported bugs using frequent terms similarities of bug attributes. Em *2011 Ninth International Conference on ICT and Knowledge Engineering*, pp. 113–117. IEEE.
- [Netto et al., 2010] Netto, F.; Barros, M. O. & Alvim, A. C. (2010). An automated approach for scheduling bug fix tasks. Em *Software Engineering (SBES), 2010 Brazilian Symposium on*, pp. 80–89. IEEE.
- [Nguyen et al., 2012] Nguyen, A. T.; Nguyen, T. T.; Nguyen, H. A. & Nguyen, T. N. (2012). Multi-layered approach for recovering links between bug reports and fixes. Em *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, FSE '12, pp. 63:1–63:11, New York, NY, USA. ACM.
- [Otoom et al., 2016] Otoom, A. F.; Al-Shdaifat, D.; Hammad, M. & Abdallah, E. E. (2016). Severity prediction of software bugs. Em *2016 7th International Conference on Information and Communication Systems (ICICS)*, pp. 92–95. IEEE.
- [Paulk et al., 1993] Paulk, M. C.; Weber, C. V.; Garcia, S. M.; Chrissis, M. B. C. & Bush, M. (1993). Key practices of the capability maturity model version 1.1.
- [Petersen et al., 2008] Petersen, K.; Feldt, R.; Mujtaba, S. & Mattsson, M. (2008). Systematic mapping studies in software engineering. *EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, pp. 68–77. ISSN 02181940.
- [Petersen et al., 2015] Petersen, K.; Vakkalanka, S. & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18. ISSN 09505849.
- [Polo et al., 1999a] Polo, M.; Piattini, M.; Ruiz, F. & Calero, C. (1999a). Mantema: a complete rigorous methodology for supporting maintenance based on the iso/iec 12207 standard. Em *Software Maintenance and Reengineering, 1999. Proceedings of the Third European Conference on*, pp. 178–181.
- [Polo et al., 1999b] Polo, M.; Piattini, M.; Ruiz, F. & Calero, C. (1999b). Roles in the maintenance process. *ACM SIGSOFT Software Engineering Notes*, 24(4):84–86. ISSN 01635948.

- [Nagwani & Verma, 2010b] Nagwani, N. K. & Verma, S. (2010b). Predictive data mining model for software bug estimation using average weighted similarity. Em *Advance Computing Conference (IACC), 2010 IEEE 2nd International*, pp. 373–378. IEEE.
- [Nagwani & Verma, 2012] Nagwani, N. K. & Verma, S. (2012). Predicting expert developers for newly reported bugs using frequent terms similarities of bug attributes. Em *2011 Ninth International Conference on ICT and Knowledge Engineering*, pp. 113–117. IEEE.
- [Netto et al., 2010] Netto, F.; Barros, M. O. & Alvim, A. C. (2010). An automated approach for scheduling bug fix tasks. Em *Software Engineering (SBES), 2010 Brazilian Symposium on*, pp. 80–89. IEEE.
- [Nguyen et al., 2012] Nguyen, A. T.; Nguyen, T. T.; Nguyen, H. A. & Nguyen, T. N. (2012). Multi-layered approach for recovering links between bug reports and fixes. Em *Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, FSE '12, pp. 63:1–63:11, New York, NY, USA. ACM.
- [Otoom et al., 2016] Otoom, A. F.; Al-Shdaifat, D.; Hammad, M. & Abdallah, E. E. (2016). Severity prediction of software bugs. Em *2016 7th International Conference on Information and Communication Systems (ICICS)*, pp. 92–95. IEEE.
- [Paulk et al., 1993] Paulk, M. C.; Weber, C. V.; Garcia, S. M.; Chrissis, M. B. C. & Bush, M. (1993). Key practices of the capability maturity model version 1.1.
- [Petersen et al., 2008] Petersen, K.; Feldt, R.; Mujtaba, S. & Mattsson, M. (2008). Systematic mapping studies in software engineering. *EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, pp. 68–77. ISSN 02181940.
- [Petersen et al., 2015] Petersen, K.; Vakkalanka, S. & Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18. ISSN 09505849.
- [Polo et al., 1999a] Polo, M.; Piattini, M.; Ruiz, F. & Calero, C. (1999a). Mantema: a complete rigorous methodology for supporting maintenance based on the iso/iec 12207 standard. Em *Software Maintenance and Reengineering, 1999. Proceedings of the Third European Conference on*, pp. 178–181.

REFERÊNCIAS BIBLIOGRÁFICAS

121

- [Prifti et al., 2011] Prifti, T.; Banerjee, S. & Cukic, B. (2011). Detecting bug duplicate reports through local references. Em *Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, p. 8. ACM.
- [Robbins & Heiberger, 2011] Robbins, N. B. & Heiberger, R. M. (2011). Plotting likert and other rating scales. Em *Proceedings of the 2011 Joint Statistical Meeting*, pp. 1058–1066.
- [Rocha et al., 2015] Rocha, H.; Oliveira, G.; Marques-Neto, H. & Valente, M. T. (2015). Nextbug: a bugzilla extension for recommending similar bugs. *Journal of Software Engineering Research and Development*, 3(1).
- [Romo & Capiluppi, 2015] Romo, B. A. & Capiluppi, A. (2015). Towards an automation of the traceability of bugs from development logs: A study based on open source software. Em *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, EASE '15, pp. 33:1–33:6, New York, NY, USA. ACM.
- [Rugg & McGeorge, 2005] Rugg, G. & McGeorge, P. (2005). The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. *Expert Systems*, 22(3):94–107.
- [Runeson et al., 2007] Runeson, P.; Alexandersson, M. & Nyholm, O. (2007). Detection of duplicate defect reports using natural language processing. Em *Proceedings of the 29th International Conference on Software Engineering*, ICSE '07, pp. 499–510, Washington, DC, USA. IEEE Computer Society.
- [Serrano & Ciordia, 2005] Serrano, N. & Ciordia, I. (2005). Bugzilla, itracker, and other bug trackers. *IEEE Software*, 22(2):11–13. ISSN 0740-7459.
- [Shokripour et al., 2012] Shokripour, R.; Kasirun, Z. M.; Zamani, S. & Anvik, J. (2012). Automatic bug assignment using information extraction methods. Em *Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on*, pp. 144–149. IEEE.
- [Singh & Chaturvedi, 2011] Singh, V. & Chaturvedi, K. K. (2011). Bug tracking and reliability assessment system (btras). *International Journal of Software Engineering and Its Applications*, 5(4):1–14.
- [Sjøberg et al., 2005] Sjøberg, D. I.; Hannay, J. E.; Hansen, O.; Kampenes, V. B.; Karahasanovic, A.; Liborg, N.-K. & Rekdal, A. C. (2005). A survey of controlled

REFERÊNCIAS BIBLIOGRÁFICAS

121

- [Polo et al., 1999b] Polo, M.; Piattini, M.; Ruiz, F. & Calero, C. (1999b). Roles in the maintenance process. *ACM SIGSOFT Software Engineering Notes*, 24(4):84–86. ISSN 01635948.
- [Prifti et al., 2011] Prifti, T.; Banerjee, S. & Cukic, B. (2011). Detecting bug duplicate reports through local references. Em *Proceedings of the 7th International Conference on Predictive Models in Software Engineering*, p. 8. ACM.
- [Robbins & Heiberger, 2011] Robbins, N. B. & Heiberger, R. M. (2011). Plotting likert and other rating scales. Em *Proceedings of the 2011 Joint Statistical Meeting*, pp. 1058–1066.
- [Rocha et al., 2015] Rocha, H.; Oliveira, G.; Marques-Neto, H. & Valente, M. T. (2015). Nextbug: a bugzilla extension for recommending similar bugs. *Journal of Software Engineering Research and Development*, 3(1).
- [Romo & Capiluppi, 2015] Romo, B. A. & Capiluppi, A. (2015). Towards an automation of the traceability of bugs from development logs: A study based on open source software. Em *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering*, EASE '15, pp. 33:1–33:6, New York, NY, USA. ACM.
- [Rugg & McGeorge, 2005] Rugg, G. & McGeorge, P. (2005). The sorting techniques: a tutorial paper on card sorts, picture sorts and item sorts. *Expert Systems*, 22(3):94–107.
- [Runeson et al., 2007] Runeson, P.; Alexandersson, M. & Nyholm, O. (2007). Detection of duplicate defect reports using natural language processing. Em *Proceedings of the 29th International Conference on Software Engineering*, ICSE '07, pp. 499–510, Washington, DC, USA. IEEE Computer Society.
- [Serrano & Ciordia, 2005] Serrano, N. & Ciordia, I. (2005). Bugzilla, itracker, and other bug trackers. *IEEE Software*, 22(2):11–13. ISSN 0740-7459.
- [Shokripour et al., 2012] Shokripour, R.; Kasirun, Z. M.; Zamani, S. & Anvik, J. (2012). Automatic bug assignment using information extraction methods. Em *Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on*, pp. 144–149. IEEE.
- [Singh & Chaturvedi, 2011] Singh, V. & Chaturvedi, K. K. (2011). Bug tracking and reliability assessment system (btras). *International Journal of Software Engineering and Its Applications*, 5(4):1–14.

- experiments in software engineering. *IEEE transactions on software engineering*, 31(9):733–753.
- [Society et al., 2014] Society, I. C.; Bourque, P. & Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*. IEEE Computer Society Press, Los Alamitos, CA, USA, 3rd edição. ISBN 0769551661, 9780769551661.
- [Soltan & Mostafa, 2016] Soltan, H. & Mostafa, S. (2016). Leanness and Agility within Maintenance Process. (January 2014).
- [Somasundaram & Murphy, 2012] Somasundaram, K. & Murphy, G. C. (2012). Automatic categorization of bug reports using latent dirichlet allocation. Em *Proceedings of the 5th India software engineering conference*, pp. 125–130. ACM.
- [Song et al., 2010a] Song, Y.; Wang, X.; Xie, T.; Zhang, L. & Mei, H. (2010a). Jdf: detecting duplicate bug reports in jazz. Em *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pp. 315–316. ACM.
- [Song et al., 2010b] Song, Y.; Wang, X.; Xie, T.; Zhang, L. & Mei, H. (2010b). Jdf: detecting duplicate bug reports in jazz. Em *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pp. 315–316. ACM.
- [Sun et al., 2011] Sun, C.; Lo, D.; Khoo, S.-C. & Jiang, J. (2011). Towards more accurate retrieval of duplicate bug reports. Em *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, pp. 253–262. IEEE Computer Society.
- [Sun et al., 2010] Sun, C.; Lo, D.; Wang, X.; Jiang, J. & Khoo, S.-C. (2010). A discriminative model approach for accurate duplicate bug report retrieval. Em *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pp. 45–54. ACM.
- [Svensson & Host, 2005] Svensson, H. & Host, M. (2005). Introducing an agile process in a software maintenance and evolution organization. Em *Ninth European Conference on Software Maintenance and Reengineering*, pp. 256–264. ISSN 1534-5351.
- [Takama & Kurosawa, 2013] Takama, Y. & Kurosawa, T. (2013). Application of monitoring support visualization to bug tracking systems. Em *Industrial Electronics (ISIE), 2013 IEEE International Symposium on*, pp. 1–5. IEEE.
- [Sjøberg et al., 2005] Sjøberg, D. I.; Hannay, J. E.; Hansen, O.; Kampenes, V. B.; Karahasanovic, A.; Liborg, N.-K. & Rekdal, A. C. (2005). A survey of controlled experiments in software engineering. *IEEE transactions on software engineering*, 31(9):733–753.
- [Society et al., 2014] Society, I. C.; Bourque, P. & Fairley, R. E. (2014). *Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0*. IEEE Computer Society Press, Los Alamitos, CA, USA, 3rd edição. ISBN 0769551661, 9780769551661.
- [Soltan & Mostafa, 2016] Soltan, H. & Mostafa, S. (2016). Leanness and Agility within Maintenance Process. (January 2014).
- [Somasundaram & Murphy, 2012] Somasundaram, K. & Murphy, G. C. (2012). Automatic categorization of bug reports using latent dirichlet allocation. Em *Proceedings of the 5th India software engineering conference*, pp. 125–130. ACM.
- [Song et al., 2010a] Song, Y.; Wang, X.; Xie, T.; Zhang, L. & Mei, H. (2010a). Jdf: detecting duplicate bug reports in jazz. Em *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pp. 315–316. ACM.
- [Song et al., 2010b] Song, Y.; Wang, X.; Xie, T.; Zhang, L. & Mei, H. (2010b). Jdf: detecting duplicate bug reports in jazz. Em *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pp. 315–316. ACM.
- [Sun et al., 2011] Sun, C.; Lo, D.; Khoo, S.-C. & Jiang, J. (2011). Towards more accurate retrieval of duplicate bug reports. Em *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, pp. 253–262. IEEE Computer Society.
- [Sun et al., 2010] Sun, C.; Lo, D.; Wang, X.; Jiang, J. & Khoo, S.-C. (2010). A discriminative model approach for accurate duplicate bug report retrieval. Em *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 1*, pp. 45–54. ACM.
- [Svensson & Host, 2005] Svensson, H. & Host, M. (2005). Introducing an agile process in a software maintenance and evolution organization. Em *Ninth European Conference on Software Maintenance and Reengineering*, pp. 256–264. ISSN 1534-5351.
- [Takama & Kurosawa, 2013] Takama, Y. & Kurosawa, T. (2013). Application of monitoring support visualization to bug tracking systems. Em *Industrial Electronics (ISIE), 2013 IEEE International Symposium on*, pp. 1–5. IEEE.