

Levantamento sobre Processo de Software Livre

Renata Pontin de Mattos Fortes, Marco Aurélio Graciotto Silva, Christian Robottom Reis

¹Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo
Av. do Trabalhador São-Carlense, 400 – Cx. Postal 668
CEP 13560-970 São Paulo, SP

{renata,magsilva}@icmc.usp.br,kiko@async.com.br

Resumo. *Este artigo apresenta alguns dos principais resultados obtidos de uma ampla pesquisa sobre processo de software livre. A pesquisa foi veiculada por meio de questionário e, com base nas respostas obtidas, puderam ser identificadas características do processo de software livre.*

Abstract. *This paper presents the main results gathered from an extensive research on Open Source Projects. The research consists of a survey and, regarding the obtained answers, characteristics of the open source process could be identified.*

1. Introdução

Compreender melhor o processo de desenvolvimento em projetos de software livre apresenta-se como um desafio, motivando o surgimento de diversas questões a respeito da organização, motivação e coordenação das equipes envolvidas. Nos últimos anos, diversas publicações científicas têm procurado descrever esses projetos de forma mais detalhada. No entanto, grande parte desses trabalhos se concentra em projetos individuais ou em aspectos particulares do processo de software.

Aparentemente, os processos de software livre contam com grupos de pessoas geograficamente distribuídas, na maioria informalmente organizados, produzindo software de alta qualidade sem auxílio de ferramentas sofisticadas. A investigação das idéias subjacentes a essa eficiência de processo de engenharia de software foi alvo de um recente trabalho desenvolvido por Reis (Reis, 2003). Sua pesquisa, reportada neste artigo, se propôs a fornecer uma ampla visão do cenário de projetos de software livre, cobrindo lacunas importantes relacionadas ao processo de software aplicado nos projetos.

Assim, o objetivo foi o de caracterizar o processo de software livre através de um amplo levantamento abordando os projetos estáveis de software livre (Reis, 2003). Foram contactados 1.102 projetos, e obtidas 519 respostas válidas (47% do total). De maneira bastante resumida, este artigo apresenta a metodologia adotada na seção 2. Os resultados obtidos são discutidos na seção 3, seguida de uma análise desses dados, na tentativa de identificar a comunidade dos processos de software de cada projeto, na seção 4. Na seção 5, são apresentadas algumas das conclusões do trabalho realizado.

2. Metodologia

Para o levantamento de dados, treze hipóteses (Reis, 2003, pp. 64-65) sobre o processo de software livre foram consideradas, obtidas através do estudo da literatura e de técnica de etnografia adotada por Reis. Os dados necessários foram obtidos através de um questionário, veiculado a pessoas com bom conhecimento específico do fluxo de trabalho e da equipe de um projeto de um software livre específico (preferencialmente o mantenedor responsável pelo projeto), e uma análise do pacote de código-fonte do projeto, a fim de comparação com os dados do questionário.

O questionário foi então testado e avaliado por alguns membros de projetos de software livre e engenheiros de software, sendo então preparada sua veiculação. Dado o desconhecimento da dimensão total da população de projetos livres a ser avaliada, optou-se por selecionar uma

amostra não-aleatória. O critério adotado foi a escolha de projetos de software livre cujo processo de software se mostrasse saudável e sustentável e, por esse motivo, projetos que tenham características de longevidade, intensa atividade ou grande e permanente destaque público. O estabelecimento da amostra, por cotas e casos críticos, envolveu a seleção de projetos em dois diretórios de software livre, o *freshmeat.net* e *sourceforge.net*, que qualificam os projetos segundo a própria comunidade de desenvolvedores.

Um importante ponto a ser discutido em qualquer levantamento é a significância da amostra. Assumindo que os sites de registro de projetos sejam representativos da população de projetos de software livre, é possível fazer uma estimativa do total de projetos de software livre observados: 27.000 projetos registrados no *freshmeat.net*, 7.019 (26%) são estáveis ou maduros; 57.000 no *sourceforge.net*, 7.231 (12,7%) como maduros ou estáveis. Assumindo, conservadoramente, que 40% dos projetos estão registrados em ambos os sites simultaneamente, é provável que a população de projetos estáveis e maduros não passe de 8.550. Com base nessa dedução, o número de respostas obtidas corresponde a até 6% do total de projetos maduros e estáveis, o que indica uma boa cobertura para o estudo empírico.

3. Resultados

A veiculação do questionário foi iniciada em setembro de 2002, se estendendo até novembro do mesmo ano. Foram enviadas mensagens para os responsáveis por projetos de software livre classificados como estáveis ou maduros e, com a chegada das respostas, fez-se a análise dos resultados e coleta secundária de dados. Os resultados quantitativos do questionário são apresentados a seguir:

Localização geográfica dos participantes - a população avaliada é internacional, com uma grande participação de norte-americanos e europeus.

Domínio de aplicação - a maioria dos projetos está relacionada à engenharia de software, redes e segurança, multimídia, navegadores e aplicações de escritório (53%).

Código-fonte - a linguagem mais utilizada é ANSI C, presente em 70,7% dos projetos analisados. Seguem sh (65,9%), Perl (39,1%), C++ (32,76%), yacc (16,76%), Python (14,26%), dentre outras. Cada projeto utiliza, em média, 3,2 linguagens. Analisando a quantidade de linha de códigos, verifica-se uma média de 60 mil linhas por projeto (KLOC). Apenas 18% dos projetos possuem mais do que a média, com apenas 10% superando 100 KLOCs.

Motivação - a maioria dos projetos (71,1%) foi iniciada por motivos pessoais, não tendo contribuição significativa de uma organização formal (13,3%) e, desde o início, feitos como software livre (58,6%). Uma parcela considerável de projetos (27,7%) teve como base um outro projeto de software.

Perfil do usuário - usuários técnicos são os principais usuários dos softwares, a começar pelo líder do projeto (75,1%) e a comunidade de software livre (62,2%), seguidos por técnicos e estudantes de computação. Usuários não técnicos figuram com 37,2% de participação.

Idade do projeto - a faixa etária dos projetos se concentra entre 2 e 5 anos (39,9%). A minoria de projetos com menos de um ano (14,6%) reflete o critério do levantamento, escolhendo projetos estáveis ou maduros.

Tamanho da equipe - equipes de desenvolvimento com menos de cinco membros predominam (74%). Projetos de dimensões maiores são responsáveis pelas equipes mais numerosas.

Modelo de liderança - encontram-se projetos cuja responsabilidade é de uma única pessoa (35,2%) e daqueles em que se observa a presença de uma organização hierárquica (57%), com delegação, formal ou informal, de responsáveis por um líder (40,3%) ou comitê (16,8%). Poucos projetos (2,5%) indicaram a ausência de uma forma de organização, sugerindo um forte consenso entre os participantes de que alguma estrutura de liderança precisa de fato existir.

Aspectos da equipe - a maioria dos membros da equipe se conhece apenas através da Internet (62,2%), com uma pequena parcela (11%) desenvolvendo de forma convencional (localmente). Observa-se também que muitos dos desenvolvedores (40,1%) executam mais de uma tarefa no projeto e que 55,5% dos desenvolvedores possuem mais de cinco anos de experiência.

Requisitos - a existência de precedentes nos projetos de software livre, constatada em *motivação*, é um facilitador de requisitos, tendo 52,8% dos projetos afirmado a replicação de softwares já existentes (35,6%) ou se basear em especificações e padrões públicos (32,2%). Além disso,

destacam-se o grande esforço na definição das funcionalidades do software (42,8%) e a independência da equipe quanto a esses requisitos, sem contribuições significativas dos usuários (40,3%).

Usabilidade - notável a baixa ênfase em engenharia de usabilidade: poucos projetos dedicam-se a boas interfaces usuário-computador. Uma pequena parcela realiza prototipação (27,7%) e estudos sobre interfaces (6,6%), com uma fraca política para controlar as alterações nas mesmas (sem necessidade de aprovação de mudanças na interface em 11,8% ou equipes específicas para este aspecto do software em 7,9%).

Documentação - muitos projetos afirmam despende um grande esforço na geração de documentação, seja ao nível de requisitos (69,6%), de projeto (30,1%) e para usuários finais (77,6%). Praticamente toda documentação é gerada inteiramente pela equipe do projeto, com apenas 11,8% dela sendo produzida por grupos externos. O índice relativamente baixo de documentação é justificado pela disponibilidade do código-livre e a possibilidade de extrair dele, seja pelos comentários ou utilização de ferramentas especializadas, os dados do projeto.

Garantia de qualidade - os softwares são testados apuradamente antes de seu lançamento (55,7%). Diferente do senso comum, não são realizadas muitas atividades de garantia de qualidade ao longo do desenvolvimento: executar testes periódicos, possuir um plano de testes (9,9%), um processo de revisão (22,9%) ou regras para integração de alterações no código (15,0%). A disponibilização do código publicamente, por meio de *releases alpha*, *beta* e *release candidates*, visando a execução de testes pelo usuário final, é a principal atividade executada para garantir qualidade (27,0%).

Ferramentas - uso intenso de gerenciamento de configuração (73,2%) e meios de comunicação simples, como listas de discussão e sites Web (ambas 66,9%) são fato na maioria dos projetos. Serviços de hospedagem, como *sourceforge.net* e *gnu.org*, também são representativos, sendo utilizados por 57% dos projetos. Outras ferramentas, como FAQs, gerenciadores de erro (como o Bugzilla) e Internet Relay Chat (IRC) são usufruídos por mais de 22% dos projetos. O baixo nível de utilização de ferramentas mais especializadas (gerenciadores de erro, editores colaborativos como *wikis* e sistemas de comunicação de tempo real) provavelmente é consequência da existência de equipes de desenvolvimentos pequenas, não justificando o esforço em mantê-las.

4. Análise dos resultados

Uma quantificação mais geral dos dados obtidos em relação à engenharia de software foi obtida pela criação de um índice denominado “Índice de Engenharia de Software”. Ele é calculado a partir de uma lista de alternativas, obtidas do questionário, consideradas relevantes para um bom processo de engenharia. A cada item dessa lista foi associado um peso, variando de 1 (esforço desprezível) a 5 (grande esforço), calculado com base em um levantamento simples com dois especialistas. De forma semelhante, foi criado um índice para descrever o uso de ferramentas de engenharia de software.

Analisando a relação entre o esforço despendido nas atividades de engenharia de software e uso das ferramentas (por meio do “Índice de Engenharia de Software”) em relação ao tamanho dos softwares (figura 1), a concentração nos valores médios para o índice de engenharia de software sugere um processo comum entre boa parte dos projetos de software livre, mas que não envolve todas as atividades abordadas no levantamento.

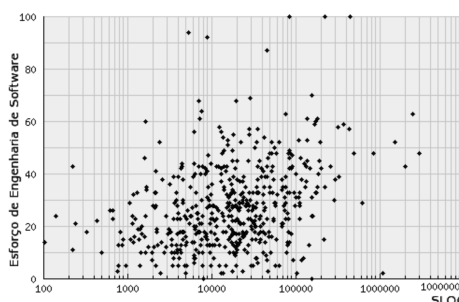


Figura 1: Gráfico *scatter* relacionando o índice de engenharia de software com o número de linhas de código de cada projeto. Fonte: (Reis, 2003, p. 130).

Analisando o índice de engenharia de software e o tamanho e idade das equipes (figura 2), verifica-se que a execução de atividades de engenharia de software está relacionada à dimensão da equipe, mas não ao tempo de existência do projeto: projetos com mais integrantes realizam mais atividades de engenharia de software. Não parece existir uma relação entre a maturidade dos projetos e utilização de novas atividades, como seria o esperado em processos de software tradicionais.

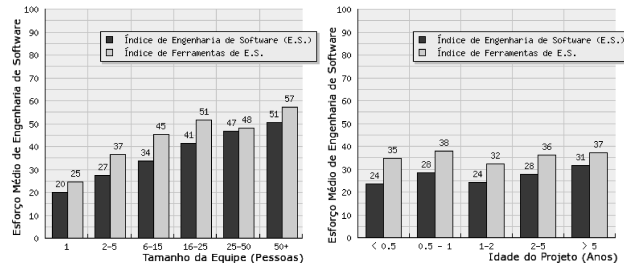


Figura 2: Gráfico representando a relação entre o índice de engenharia de software, o tamanho da equipe e a idade do projeto. Fonte: (Reis, 2003, p. 131)

Define-se, então, um ciclo de vida para software livre, cada projeto instanciando-o sob influência de sua cultura interna, histórico e objetivos. O ciclo é composto por quatro fases: criação, lançamento público, crescimento e organização, e maturidade. As fases mais importantes são as que procedem ao lançamento público: a partir desse momento, o projeto entra em processo de evolução contínua.

5. Conclusão

O levantamento realizado permitiu visualizar interessantes características do processo de software livre, algumas inclusive fugindo ao senso comum, tal como a relativamente baixa ocorrência de atividades de garantia de qualidade. Isso contrapõe o fato de softwares livres apresentarem qualidade.

Uma justificativa são os aspectos humanos que existem nos projetos de software livre: a forte interação usuário-desenvolvedor, o trabalho auto-gratificante. No mesmo caso das atividades de teste, provavelmente esses mecanismos sociais e psicológicos têm impacto significativo na qualidade do software. Conforme exemplificado por D. J. Bernstein: “*Reliability means never having to say you’re sorry*”.

Os projetos livres oferecem uma chance de cooperar com pessoas com quem não se teria contato normalmente, trabalhando para um objetivo prático e com resultados concretos. Essa é característica chave de todo processo de software livre.

Referências

Reis, C. R. (2003). Caracterização de um processo de software para projetos de software livre. Dissertação de Mestrado, Universidade de São Paulo, São Carlos - SP.