

Controla: Ferramenta de Apoio ao Processo de Desenvolvimento de Software em pequenas empresas

Clayton Vieira Fraga Filho¹, José Maurício dos Reis²

^{1,2}Faculdade de Viçosa, Rua Dr. Milton Bandeira, Viçosa Shopping, CEP 36.570-000, Viçosa-MG, Brasil.
{claytonfraga, mauricio1968}@gmail.com

Resumo - O presente trabalho tem como propósito apresentar o CONTROLA, uma ferramenta de apoio ao processo de desenvolvimento de Software para pequenas empresas, com foco no gerenciamento de requisitos. Permite a identificação dos requisitos junto ao *stakeholder* (cliente/usuário do sistema), seu detalhamento e o gerenciamento das mudanças através das matrizes de rastreabilidade e controle de versões. A ferramenta está sendo utilizada por usuários em todo o Brasil, dentre eles estudantes de graduação e pós-graduação, professores e pequenas empresas de desenvolvimento de software.

Palavras-chave: Engenharia de Software; Processos de Desenvolvimento de Software; Gerência de Requisitos; Gerência de Mudanças; Controla.

Abstract - The present work has as purpose to present CONTROLA, a support tool for software development process for small companies, with focus in the administration of requirements. It allows the identification of the requirements with the stakeholder (client/user of the system), its details and the administration of the changes through the traceability matrixes and version control. The tool this being used by users throughout Brazil, among them graduation students and masters degree, teachers and small development companies of software.

Key-words: Software Engineering; Software Development Processes; Requirements Management; Change Management; Controla.

Introdução

Com a globalização, os mercados tornaram-se mais competitivos e a demanda por produtos e serviços de melhor qualidade surge como fator decisivo no momento da escolha do parceiro de negócios nos dias de hoje. No âmbito do desenvolvimento de Software não é diferente, o cliente, cada vez mais exigente, demanda por um sistema dentro do escopo solicitado e que atenda suas necessidades, desenvolvido no menor tempo possível.

Na indústria, em geral, a qualidade de um produto depende diretamente dos processos, materiais e técnicas utilizadas, portanto, é importante que as empresas desenvolvedoras de Software tenham em foco que a definição de um processo de desenvolvimento, a utilização de ferramentas e técnicas adequadas, garante melhor desempenho e qualidade do produto final e, conseqüentemente, maior satisfação do cliente [1].

A construção e implantação de um processo de desenvolvimento de Software nas pequenas empresas desenvolvedoras não é algo comum, pois geralmente o foco está no desenvolvimento para o atendimento imediato do

problema do cliente. Aos requisitos estão associados os principais problemas do desenvolvimento de software [2], sendo assim, o gerenciamento dos requisitos e das mudanças ao longo do projeto torna-se um processo bastante dispendioso, uma vez que, não há documentação formal e padronizada das necessidades do cliente. Esse quadro pode ser agravado pela grande rotatividade de mão-de-obra (*turnover*). Frequentemente encontramos empresas com sérios problemas em atender seus clientes no prazo e no escopo do produto, por que no decorrer do projeto um dos membros da equipe deixa a empresa, levando consigo o conhecimento das regras de negócio e da especificação do Software. O tempo e os custos para recuperação desse conhecimento podem ser muito altos dependendo da fase em que o projeto se encontra.

Além das questões citadas acima, a falta de registro formal e padronizado de requisitos afeta o processo de manutenção adaptativa ou evolutiva do Software, dificultando a identificação dos impactos da mudança de algum requisito sobre o restante do sistema, potencializando a ocorrência de erros.

O registro e o gerenciamento das solicitações do cliente e as suas mudanças ao longo ciclo de vida do projeto exigem, da empresa desenvolvedora de Software, organização e padronização, para que o produto final apresente qualidade, proporcionando uma manutenção mais segura e a satisfação do cliente.

Metodologia

A metodologia aplicada no desenvolvimento do Controla é resultado de estudos comparativos realizados nas principais metodologias hoje existentes como Praxis [3], recomendações do PMBoK [4] e SWEBoK [5], sempre tendo como foco, pequenas empresas de desenvolvimento de software. Estas organizações não contam com recursos para investir na implantação de um processo de desenvolvimento complexo e ferramentas de apoio disponíveis no mercado. Procurou-se definir um conjunto básico de procedimentos que pudesse ser adaptado à realidade dessas organizações.

Definiu-se um conjunto básico de artefatos, contemplando o processo identificação das expectativas dos usuários, identificação e especificação de requisitos, projeto, testes, identificação de erros, gerenciamento das implementações, contemplando assim todo o ciclo de desenvolvimento de software, desde a identificação do escopo, projeto, implementação, testes, passando pelo plano de liberação de software, gerência de mudanças, permitindo a rastreabilidade entre todos os artefatos presentes neste ciclo. A matriz de rastreabilidade é uma importante ferramenta para a equipe de projeto, pois permite visualizar de forma gráfica a relação entre os artefatos e eventuais suspeições que podem surgir durante o desenvolvimento de um projeto de software.

A ferramenta oferece a possibilidade de:

1. Levantamento e registro das expectativas (visão) e requisitos dos usuários;
2. Projeto e especificação do sistema por meio de Casos de Uso;
3. Controle e acompanhamento das implementações identificadas e cadastradas para um sistema;
4. Registro dos usuários, membros de equipe de desenvolvimento e clientes do sistema, bem como sua hierarquia para solicitação e aprovação de requisitos;
5. Levantamento das regras de negócio, de forma que possam ser verificáveis;
6. Validação dos requisitos junto ao cliente;
7. Ferramentas de estimativa de prazo do projeto baseada em Casos de Uso;
8. Ferramenta de Priorização de Requisitos fundamentada na participação ativa do usuário;
9. Agrupamento dos requisitos por tipos;
10. Especificação e Detalhamento dos requisitos;

11. Relacionamentos de dependência entre os requisitos para os diferentes grupos de artefatos por meio de Matrizes de Rastreabilidade;

12. Definição de Níveis de importância dos requisitos para os usuários;

13. Definição dos Requisitos não-funcionais;

14. Criação e Registro dos Casos de Testes para um Caso de Uso;

15. Relacionamento dos Casos de Uso que atendem determinado requisito;

16. Registro dos resultados dos testes e correções para erros encontrados nas fases de teste ou operação do sistema;

17. Registro do histórico de alterações nos requisitos;

18. Avaliação do impacto de alterações no sistema por meio de rastreabilidade;

19. Definição de tipo de implementações;

20. Registro de Implementações por Casos de Uso de Software;

21. Visualização de itens suspeitos nas matrizes;

22. Detalhamento das implementações em pseudocódigo por meio de sua descrição.

22.1. Consulta impressão e visualização de Indicadores (estatísticas) para os artefatos do software;

23. Emissão de Relatórios:

23.1 Plano de Projeto;

23.2. Documento de Requisitos;

23.3 Documento de Casos de Uso;

23.4. Resultados das matrizes de rastreabilidade;

23.5. Resultados das ferramentas de simulação.

24. Exportação de Dados para XML, Bitmap e CSV;

25. Impressão de Relatórios para diversos formatos;

O relacionamento entre os artefatos do sistema pode ser visualizado na figura 1.

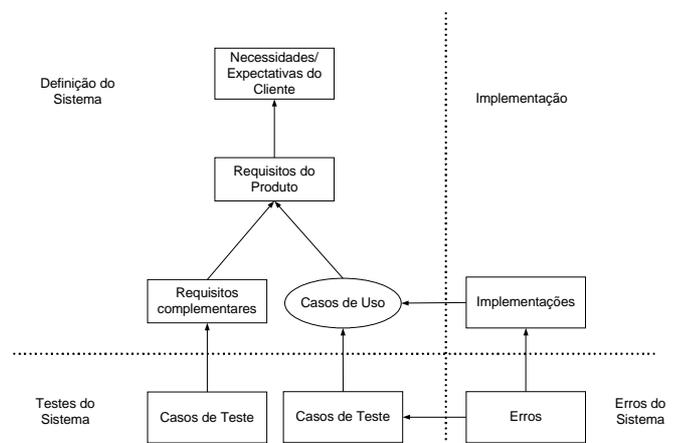


Figura 1 – Relacionamento entre artefatos do sistema em seu ambiente.

A ferramenta ainda oferece à equipe de projeto a possibilidade de criar cenários usando as ferramentas de simulação, permitindo a manipulação das variáveis, impressão e troca de arquivos com outras equipes de projeto. Os modelos implementados são 2, o primeiro é o Método de priorização de requisitos baseada em valor, custo e risco [6]. O outro modelo é destinado a estimativa de esforço e tamanho de projetos, denominado Método de Pontos por Caso de Uso (UCPM) [7]. Ambos são úteis, pois oferecem maior grau de certeza nos procedimentos realizados na estimativa e gerência do projeto.

A figura 2 apresenta as funcionalidades do Controla.

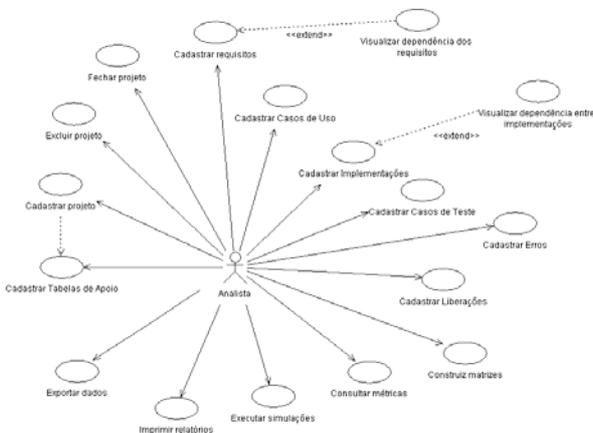


Figura 2 – Diagrama de Casos de Uso do sistema Controla.

Resultados

O foco inicial do Controla são as pequenas empresas e desenvolvedores autônomos. É sabido que este público, talvez pela falta de recursos e até mesmo de tempo, não têm condições de agregar qualidade ao seu processo produtivo ou adquirir ferramentas que os auxiliem, sendo assim, o software contempla diversas etapas do processo de qualidade, oferecendo um conjunto de funcionalidades, dentre elas:

- Permite o cadastro e manutenção de projetos de software;
- Gerenciamento dos Requisitos cadastrados para um projeto;
- Gerenciamento de Casos de Uso;
- Gerenciamento de Casos de Teste e Erros;
- Planejamento de Liberações de software para o cliente;
- Gerenciamento de Implementações;
- Controle de Dependência entre implementações;

Matriz de Rastreabilidade (Traceability Matrix) para os artefatos contemplados pelo Controla:

- Rastreabilidade Requisitos - Casos de Uso;
- Rastreabilidade Casos de Uso - Implementações;
- Rastreabilidade Casos de Uso - Casos de Teste;
- Rastreabilidade Casos de Teste - Erros;
- Rastreabilidade Liberações - Casos de Uso;
- Rastreabilidade Liberações - Casos de Teste;
- Rastreabilidade Erros - Liberações;
- Ferramenta de estimativa de tamanho de software por Pontos de Casos de Uso;
- Ferramenta para priorização de Requisitos [6];
- Emissão de documentos de Plano de Projeto; Casos de Uso; Especificação de Requisitos e Exportação de dados;

O Controla conta hoje com usuários em todo o país. Cerca de 4400 cópias foram distribuídas após a divulgação em importantes Web sites especializados, jornais e revistas eletrônicas.

O resultado da divulgação foi muito amplo, pesquisadores, empresas, órgãos governamentais, estudantes, professores entraram em contato fornecendo sugestões e descrevendo como o software está sendo útil no seu cotidiano.

Dentre as principais aplicações do Controla, percebidas pelo retorno dos usuários, podemos citar principalmente:

- Como ferramenta de ensino da Disciplina Engenharia de Software;
- Como ferramenta de Apoio ao Processo de Desenvolvimento de Software em pequenas e médias empresas;
- Como ferramenta de pesquisa: despertou interesse de pesquisadores que já estão em contato com o Controla;

A figura 3 apresenta a interface principal do Controla.



Figura 3 – Interface principal do software Controla.

Discussão e Conclusões

O Controla está sendo utilizado como ferramenta de apoio ao processo de desenvolvimento de software em alguns projetos e incorporando melhorias e ajustes contínuos. A partir deste semestre será utilizada como ferramenta acadêmica na disciplina Engenharia de Software na Faculdade de Viçosa – MG. A versão 1.0 oferece muitos recursos para pequenas empresas e equipes que não possuem metodologias definidas ou não dispõem de recursos para adquirir as poderosas ferramentas CASE disponíveis no mercado. Uma versão multi-usuário para ambiente Web está sendo desenvolvida. Possibilitará a colaboração on-line para equipes de desenvolvimento e grupos de stakeholders em múltiplos projetos, incluindo recursos de fórum de discussão, workflow e acompanhamento de métricas pré-estabelecidas.

Referências

1. Kan, Stephen H., 2002, Metrics and Models in Software Quality Engineering, Second Edition, Addison Wesley, Boston, EUA.
2. Blaschek, José Roberto, 2002, Gerência de Requisitos: O principal problema dos projetos de Software, Disponível em <<http://www.bfpug.com.br/islig-rio/Downloads/Gerencia%20de%20Requisitos-o%20Principal%20Problema%20dos%20Projetos%20de%20SW.pdf>> (06/05/2005).
3. Paula Filho, Wilson de Pádua. Engenharia de Software Fundamentos, Métodos e Padrões. LTC - Editora, 2002.
4. Pmi Standards Committee, A Guide to the Project Management Body of Knowledge. Upper Darby, PA, PMI Management Institute, 1996. Disponível em: <<http://www.pmi.org/publictn/pmbok>> (06/04/2005).
5. Abran, Alain et al, SWEBOK - Guide to the Software Engineering Body of Knowledge, IEEE Computer Society, 2004.
6. Wiegers, Karl E., 2003, Software Requirements, Second Edition, Microsoft Press.
7. Kusumoto, Shinji, Jean Jéferson, 2005, Effort Estimation Tool Based on Use Case Points Method, Disponível em <<http://www.spars.info/~lab-db/betuzuri/archive/462/462.pdf>>.