

Uso de Ferramentas Open Source para gestão de Projetos

Eduardo Cardoso Moraes

Programa de Pós-Graduação em Ciência da Computação – Centro de Informática –

Universidade Federal do Pernambuco

Resumo. Atualmente um importante desafio em aberto da engenharia de software é o bom gerenciamento de projetos de software. O gerenciamento de projetos de software abrange todas as fases a que o software desde sua concepção até sua validação pelo cliente, assim, faz-se necessário que ferramentas sejam usadas para automatizar esse processo. Nesse contexto, o bom gerenciamento do software está diretamente ligado a escolha de boas ferramentas para subsidiar decisões e manter a consistência dos processos durante o tempo de desenvolvimento do software. Este trabalho apresenta uma comparação qualitativa entre ferramentas open source, e descreve com maiores detalhes de três ferramentas: Achievo, Project Pier e Red Mine, utilizando como critérios de comparação o documento de processos para gerência de projetos descrito MPS-BR (nível G).

1. Introdução

A engenharia de software segundo o IEEE, apud Pressman, pode ser definida como “(1) a aplicação de uma abordagem sistemática, disciplinada e quantificável, para desenvolvimento, operação e manutenção do software. (2) Ainda segundo Pressman(PRESSMAN, 2006), define que a engenharia de software é uma tecnologia em camadas (Ferramentas, Métodos, Processo, Foco na qualidade) e a organização deve se apoiar num compromisso de qualidade.

O gerenciamento de projetos de software representa a primeira camada do processo de engenharia de software. O gerenciamento de projetos compreende atividades que envolvem medições, estimativas, análise de erros, programação de atividades, monitoramento e controle. É a aplicação de conhecimentos, habilidades, ferramentas e técnicas em projetos com o objetivo de atingir ou até mesmo exceder as necessidades e expectativas dos clientes e demais partes interessadas do projeto. A gerência de projetos de software se caracteriza por tratar sobre um produto intangível, muito flexível e com processo de desenvolvimento com baixa padronização.

Gerenciar todas as fases do desenvolvimento de um software manualmente, sem o apoio de ferramentas pode se tornar uma tarefa árdua e levar a uma série de erros, além de exigir um tempo considerável do projeto. Para resolver esse problema, têm-se algumas ferramentas que automatizam

uma série de atividades rotineiras ao gerenciamento do software. Não obstante, é importante se verificar a qualidade dessas ferramentas e seu nível de completude, eficiência e usabilidade.

Este trabalho tem como objetivo comparar/avaliar ferramentas de gerência de projetos de Software open source observando características funcionais na tentativa de fornecer uma visão crítica sobre a qualidade e completude de tais ferramentas, baseando-se em algumas normas do programa para Melhoria de Processos do Software Brasileiro - MPS-BR [MPS.BR 2009], nível G no tocante a processos para gerência de projetos. Isso pode ser útil, pois fornece encaminhamento para escolha de uma das ferramentas avaliadas ou mesmo pode facilitar a análise de outras ferramentas.

As ferramentas escolhidas para a análise foram Achievo[ACHIEVO 2010], Project Pier [PROJECT-PIER 2010] e Red Mine [REDMINE 2010] .

O restante desse trabalho está estruturado da seguinte maneira. A seção 2 apresenta uma visão geral sobre as normas no MPS-BR no que se refere à gerência de projetos, sendo esta a norma aos qual este trabalho foi baseado para a confecção de critérios de avaliação utilizados. A seção 3 apresenta as três ferramentas que serão avaliadas nesse trabalho.

A seção 4 define os critérios do MPS-BR utilizados para avaliação das ferramentas observando-se características funcionais, além de descrever a metodologia aplicada para comparação. Por fim, a seção 5 discute os resultados obtidos na avaliação das ferramentas.

2. Contextualização

Com o intuito de informar e subsidiar os gestores de projetos sobre qual ferramenta utilizar em seus projetos, faz-se necessária uma comparação clara e que abranja critérios úteis e importantes. Antes de iniciarmos o nosso uso é relevante que seja feito um estudo visando à definição detalhada dos principais requisitos que tais ferramentas devem cumprir antes de iniciar a comparação propriamente dita. Em especial, quando se fala de gerencia de projetos de software, existem vários modelos que definem normas para uma maior qualidade no processo de software. Este trabalho fará uso de algumas normas do programa para Melhoria de Processos do Software Brasileiro - MPS-BR [MPS.BR 2009], nível G, processos para gerência de projetos.

O MPS.BR ou Melhoria de Processos do Software Brasileiro é simultaneamente um movimento para a melhoria da qualidade (Programa MPS.BR) e um modelo de qualidade de processo (Modelo MPS). Ele é baseado nas normas ISO/IEC 12207 e ISO/IEC 15504 e na realidade do mercado brasileiro, bem como é compatível com o CMMI.

No Brasil, existe incentivo a sua adoção e uma das principais vantagens do modelo é seu custo reduzido de certificação em relação as normas estrangeiras, sendo ideal para micro, pequenas e médias empresas.

O MPS-BR tem como objetivo definir um modelo de melhoria e avaliação de processo de software, preferencialmente para as micro, pequenas e médias empresas, de forma a atender as suas necessidades de negócio e a ser reconhecido nacional e internacionalmente como um modelo aplicável à indústria de software. O Modelo de Referência MPS-BR define níveis de maturidade que são uma combinação entre processos e capacidade de processos.

Os níveis de maturidade estabelecem patamares de evolução de processos, caracterizando estágios de melhoria de implementação de processos na organização. O nível de maturidade em que se encontra uma organização permite prever seu desempenho futuro em uma ou mais disciplinas. O MPS-BR define sete níveis de maturidade: A (Em Otimização), B (Gerenciado Quantitativamente), C (Definido), D (Largamente Definido), E (Parcialmente Definido), F (Gerenciado) e G (Parcialmente Gerenciado). A escala de maturidade se inicia no nível G e progride até o nível A.

Em particular, estaremos interessados no nível G do MPS-BR. O nível G é composto pelos processos de Gerência de Projeto e Gerência de Requisitos satisfazendo os atributos de processo AP 1.1 e AP 2.1. O propósito do Processo Gerência de Projetos (GPR) é identificar, estabelecer, coordenar e monitorar as atividades, tarefas e recursos que um projeto necessita para produzir um produto e/ou serviço, no contexto dos requisitos e restrições do projeto.

Com isso, usaremos os resultados esperados definidos pelas GPRs do nível G do MPS-BR para construir um conjunto de critérios que serão utilizados na avaliação da completude funcional das ferramentas.

3. Ferramentas de gerência de projeto em estudo

Esta seção descreve em linhas gerais as ferramentas utilizadas na comparação proposta por esse trabalho: Achievo, Redmine e ProjectPier.

3.1. Achievo

Achievo é um software de gestão flexível baseada na Web, divulgando que tem seu foco e é apropriada para empresa de porte médio que precisa manter o controle de seus recursos, projetos, clientes, contatos, planejamento e programação diária. A ferramenta Achievo está disponível em mais de 20 idiomas. O sistema é dividido em cinco módulos que são:

Registro de Tempo: A parte de registro de tempo é a parte onde os funcionários podem registrar o tempo de projeto / fase combinações de atividade. Isso inclui o registro diário ou registro semanal, horas de relatórios e bloqueio de períodos de tempo.

O gerenciamento de projetos:Esta parte inclui a gestão de projetos, fases, tarefas, notas, estatísticas, planejamento, os membros do projeto.

Customer Relationship Management:A seção de CRM Achievo permite administrar seus fornecedores, prospects e clientes e para cada uma dessas pessoas a contactar. Também inclui uma parte da gestão dos contratos em que os contratos podem ser gerenciados por relação. Projetos também são gerenciados pela relação.

Gestão de Recursos Humanos: Na seção de gestão de RH, a gestão dos funcionários e dos contratos dos funcionários residem. Contrato horas e perfis de segurança também podem ser tratadas aqui.

Agendamento:agendador de Achievo permite aos usuários compartilhar uma agenda com os outros usuários. O escalonador inclui visualização diária, Visão semanal, MonthView, eventos recorrentes e os avisos de e-mail. Além disso, é possível comparar os horários.

The screenshot displays the Achievo web application interface. At the top, there is a navigation bar with links for 'Pim', 'User preferences', and 'Logout'. The user is logged in as 'manager [test]'. The main content area is titled 'Master Gantt' and features a Gantt chart for 'Project A'. The chart shows a timeline from January 1, 2010, to March 12, 2010, with weekly intervals labeled w1 through w11. The chart is currently in 'Planned' mode. A sidebar on the left contains a menu with categories like 'Time registration', 'Reports', 'Projects', 'Sales', 'Organizations', 'Employees', 'Setup', and 'Support'. The 'Projects' section is expanded, showing options like 'Select project:', 'No project selected', and various project management tools.

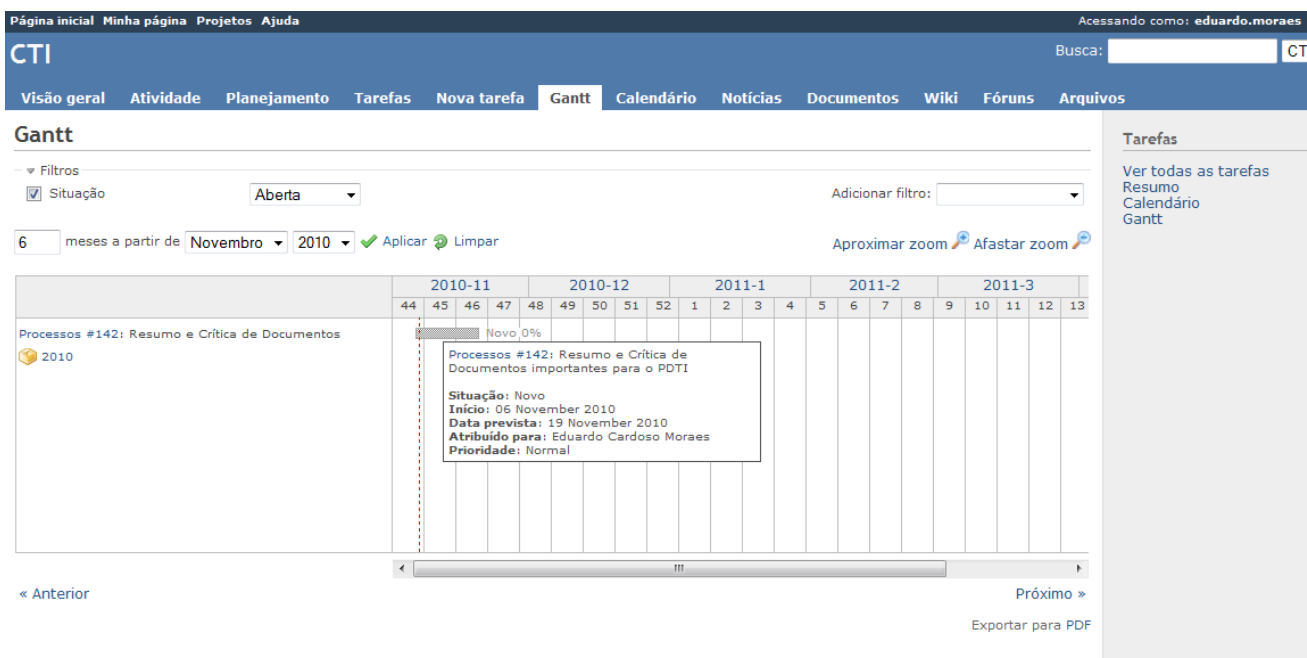
3.2. Redmine

O Redmine [REDMINE 2010] é uma ferramenta de gerência de projetos Open Source implementado em Ruby, utilizando o framework Ruby on Rails. Isso a torna portátil, ou seja, a ferramenta roda em sistemas GNU/Linux, Windows, Unix e Mac, direto no browser. Algumas características importantes incluem sua facilidade de instalação, sistema de autenticação, gerencia de tarefas, etc. O sistema tem um suporte ao usuário de alta qualidade, tem um wiki onde tem-se disponível um guia completo sobre a ferramenta, desde a parte de instalação até a parte de migração, além de um tutorial completo sobre as funcionalidades do Redmine. Possui também uma coleção de plugins para aumentar a eficiência do sistema. Esses plugins podem ser instalados no próprio programa na parte repositórios.

No Redmine é possível adicionar uma nova tarefa com as seguintes características: status(nova, completa, etc.), prioridade (normal, alta, urgente, etc.), responsável pela execução da tarefa, data de início, data de término, horas estimadas até a conclusão da tarefa e a porcentagem de execução da tarefa. No decorrer do projeto é possível acompanhar e editar o andamento da atividade. A ferramenta possui gráfico de Gantt e calendário para acompanhar das atividades, ou seja, é possível acompanhar o cronograma do projeto através de calendário gráfico que marca o dia de começo do projeto e de término. Porém, não existe um campo específico para se estabelecer os recursos necessários, nem o perfil do membro da equipe, e ainda, só é possível atribuir uma tarefa a um único membro da equipe, além de não poder determinar fases do projeto, ou seja, só é possível dividir o projeto em tarefas individuais.

A ferramenta possui um sistema de controle de versões que permite a manutenção de um histórico de mudança nos arquivos de um projeto. Uma seção interessante da ferramenta é a seção "news", onde se pode colocar notícias sobre o projeto, relatar bugs e até riscos, mas não possui uma parte específica para estimar riscos ou para fazer acompanhamento dos mesmos, nem calcular impacto ou probabilidade de ocorrência. O gerenciamento de recursos humanos é insuficiente, pois só é possível adicionar um usuário, atribuir uma tarefa e a função que o membro vai desempenhar na mesma, porém não há como descrever características e habilidades do usuário.

O Redmine não possui estimativa de custo para cada fase ou atividade nem orçamento para o projeto todo. Nem é possível usar dados históricos como base para estimativas para o projeto.



3.3. Project Pier

ProjectPier [PROJECT-PIER 2010] é uma aplicação escrita na linguagem PHP, software livre, criada para gerenciar tarefas, projetos e equipes através de uma

intuitiva interface web. O ProjectPier é um fork da ferramenta de colaboração online activeCollab, está escrito em PHP5 e baseado em uma base de dados MySQL. Como funciona na web, é um produto multiplataforma, e além disso está disponível em 24 idiomas, garantindo uma disponibilidade muito alta. Sua licença de software livre é a Affero General Public License.

ProjectPier é bastante diferente de outros software tradicionais de gerencia de projetos: a plataforma permite que equipes pequenas troquem mensagens, listas de tarefas e arquivos rapidamente. Sua instalação é bastante simplificada e automatizada por uma interface web disponibilizada pelo aplicativo. É preciso ter um servidor web configurado, habilitado para execução de scripts PHP, além de um banco de dados. Entre as suas características principais destacam:

- _ Fácil de instalar;
- _ Baseado na web: versatilidade, compatibilidade, conforto e potencia da web.
- _ Organizado: o Project Pier organiza os projetos em pequenas metas, mensagens, listas de tarefas e arquivos.
- _ Confiável: ProjectPier está baseado em tecnologias de código aberto estabelecidas, maduras e confiáveis - PHP5 e MySQL. É rápido, estável e seguro.
- _ Comunidade: ProjectPier é desenvolvido por uma comunidade ativa de usuários que compartilham experiências de milhares de projetos de todo o mundo.
- _ A característica multicliente permite gerenciar projetos de diferentes clientes de maneira simultânea.

Existem neste software a gestão de direitos em 3 níveis.

- Administrador
- Usuário da companhia proprietária.
- Usuário da companhia cliente.

_ Privacidade: As mensagens, tarefas, metas, arquivos e comentários se podem marcar como privados, ocultando informação crítica dos clientes.

Apesar da facilidade encontrada na instalação da ferramenta, algumas deficiências ficaram claras ao longo do uso. Alguns pontos observados envolvem desde a falta de representações visuais das fases do projeto, por exemplo não possui gráfico de Gantt, até características de planejamento de recursos. Ainda, notou-se uma fraca documentação das funcionalidades da ferramenta, dificultando a usabilidade do software.

Em resumo, o ProjectPier é uma iniciativa muito interessante, fácil de instalar, Facilidade de encontrar seus recursos, visualmente limpa e com características que podem satisfazer muitas necessidades básicas de gerencia de projetos. Mas um gestor com necessidades mais avançadas não vai se satisfazer com o que esta ferramenta apresenta.

4. Metodologia

Para avaliar as ferramentas foram usadas os documentos do MPS-BR Guia Geral, MPS-BR Guia de Implementação - Parte 1: Nível G e NBR ISOIEC-9126. A partir disto foram listadas um conjunto de características relevantes e verificado se possui cada uma.

Nas subseções seguintes, temos definida uma lista com requisitos funcionais e não funcionais para um software de gerenciamento de projetos. Essa lista será utilizada na comparação das ferramentas, conforme metodologia descrita abaixo.

4.1. Requisitos

Deve ser analisado se o software tem ou não tem o requisito de acordo com o MPS-BR nível G.

GPR 1. O escopo do trabalho para o projeto é definido;

GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados;

GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos;

GPR 4 O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas;

GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos;

GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados;

GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo;

GPR 8. Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados;

GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los, incluindo, se pertinente, questões de privacidade e segurança;

GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos;

GPR 11. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados;

GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido;

GPR 13. O projeto é gerenciado utilizando-se o Plano do Projeto e outros planos que afetam o projeto e os resultados são documentados;

GPR 14. O envolvimento das partes interessadas no projeto é gerenciado;

GPR 15. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento;

GPR 16. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas;

GPR 17. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão;

4.2. Metodologia

Os requisitos foram elencados e as ferramentas configuradas e analisadas conforme a tabela a seguir:

| REQUISITO MPS-BR NÍVEL G | ACHIEVO | REDMINE | PROJECT PIER |
|---|----------------|----------------|---------------------|
| GPR 1. O escopo do trabalho para o projeto é definido; | X | X | X |
| GPR 2. As tarefas e os produtos de trabalho do projeto são dimensionados utilizando métodos apropriados; | X | X | X |
| GPR 3. O modelo e as fases do ciclo de vida do projeto são definidos; | X | X | X |
| GPR 4 O esforço e o custo para a execução das tarefas e dos produtos de trabalho são estimados com base em dados históricos ou referências técnicas; | X | X | X |
| GPR 5. O orçamento e o cronograma do projeto, incluindo a definição de marcos e pontos de controle, são estabelecidos e mantidos; | X | X | X |

| | | | |
|--|--------------|--------------|--------------|
| GPR 6. Os riscos do projeto são identificados e o seu impacto, probabilidade de ocorrência e prioridade de tratamento são determinados e documentados; | NÃO | PARCIALMENTE | NÃO |
| GPR 7. Os recursos humanos para o projeto são planejados considerando o perfil e o conhecimento necessários para executá-lo; | X | X | X |
| GPR 8. Os recursos e o ambiente de trabalho necessários para executar o projeto são planejados; | X | X | X |
| GPR 9. Os dados relevantes do projeto são identificados e planejados quanto à forma de coleta, armazenamento e distribuição. Um mecanismo é estabelecido para acessá-los; | PARCIALMENTE | X | PARCIALMENTE |
| GPR 10. Um plano geral para a execução do projeto é estabelecido com a integração de planos específicos; | X | X | |
| GPR 11. A viabilidade de atingir as metas do projeto, considerando as restrições e os recursos disponíveis, é avaliada. Se necessário, ajustes são realizados; | NÃO | X | X |
| GPR 12. O Plano do Projeto é revisado com todos os interessados e o compromisso com ele é obtido; | NÃO | X | X |
| GPR 13. O projeto é gerenciado utilizando-se o Plano do Projeto e outros planos que afetam o projeto e os resultados são documentados; | X | X | NÃO |
| GPR 14. O envolvimento das partes interessadas no projeto é gerenciado; | PARCIALMENTE | X | X |
| GPR 15. Revisões são realizadas em marcos do projeto e conforme estabelecido no planejamento; | X | X | X |
| GPR 16. Registros de problemas identificados e o resultado da análise de questões pertinentes, incluindo dependências críticas, são estabelecidos e tratados com as partes interessadas; | X | PARCIALMENTE | X |
| GPR 17. Ações para corrigir desvios em relação ao planejado e para prevenir a repetição dos problemas identificados são estabelecidas, implementadas e acompanhadas até a sua conclusão; | NÃO | NÃO | NÃO |

5. Conclusão

A partir deste estudo, da aplicação da metodologia, avaliação e comparação das ferramentas, pode-se inferir que não existem diferenças significativas entre as ferramentas. Porém a ferramenta Redmine se apresenta como a ferramenta

mais completa e estável, além de possuir uma grande comunidade que a mantém.

Todas têm numerosas possibilidades de melhoria quanto às necessidades de uma organização, mas não deve-se esquecer de que são ferramentas ainda imaturas e em constante evolução e desenvolvimento.

Entre as deficiências principais detectadas em todas ferramentas destacamos as seguintes:

- Não existe a possibilidade de estimar esforços e custos necessários das tarefas de projeto, assim como seu orçamento, para um futuro cruzamento de dados.
- Em relação aos riscos, embora for possível defini-los, não é possível determinar o seu impacto, probabilidade de ocorrência e prioridade de tratamento
- Não existem mecanismos para avaliar a viabilidade de atingir as metas do projeto.

Enfim, embora sejam todas elas boas ferramentas gratuitas e livres, para que os gestores possam se beneficiar com a economia financeira(pois não precisam adquirir licenças), devem refletir bem sobre a necessidade de suporte, porque ainda existe pouca documentação e falta ainda número maior de publicações acerca do tema. Devido a essas soluções evoluírem constantemente é importante que o gestor esteja atento a esses comparativos entre soluções proprietárias e livres, assim como declarações sobre experiências e vantagens e desvantagens de cada uma.

Referência bibliográfica

ACHIEVO(2010). Achievo. <http://www.achievo.com>, Ultimo acesso em outubro de 2010.

BROWN, ALAN W., On Components and Objects: The Foundation of Component- Based Development, Assessment of Software Tools and Tecnology, Procedings Fifth International Symposium on Procedings - IEEE, 1997.

MPS.BR (2009). Melhoria de processo do software brasileiro - guia geral. Disponível em <http://www.softex.br>. Ultimo acesso em outubro de 2010.

PRESSMAN, ROGER S., *Engenharia de Software- (6ª edição)*, São Paulo, Ed. McGrawHill, 2006.

PROJECT-PIER (2009). Project pier. <http://www.projectpier.org/>. Ultimo acesso em outubro de 2010.

PETERS, JAMES F., *Engenharia de Software: Teoria e Prática*, Rio de Janeiro, Editora Campus, 2001.

REDMINE (2009). Redmine. <http://www.redmine.org>. Ultimo acesso em outubro de 2010.

SOMMERVILLE, I. Software Engineering (International Computer Science Series). 5a Edição. Reading: Addison-Wesley, 1995.

SWEBOK 2004, Guide for the Software Engineering Body of Knowledge, 2004 version, IEEE Computer Society, California, EUA.