Desenvolvimento Colaborativo e Integrado de Sistemas: Caso de Sucesso no CPD-UFRGS

Henrique Dias, Nadjia Feller, Thiago Motta
Centro de Processamento de Dados
Universidade Federal do Rio Grande do Sul
{hdpsantos,nadjia.feller,thiago}@cpd.ufrgs.br

RESUMO
O aumento da quantidade de aplicações desenvolvidas no CPD da UFRGS exigiu, nos últimos anos, a utilização de ferramentas mais aderentes ao desenvolvimento de software para manter a qualidade dos sistemas entregues à comunidade. Gerência de projetos, versionamento de código, componentes compartilhados, banco de dados centralizado entre outros processos de trabalho ajudam na melhoria da qualidade dos sistemas. Neste artigo serão apresentadas as ferramentas e processos que ajudaram na ampliação e organização do processo de desenvolvimento de sistemas de informação no CPD.

Palavras-chave: Desenvolvimento de aplicações, gerenciamento de projetos, qualidade de software.

ABSTRACT
The increase of applications developed in CPD UFRGS demanded, in recent years, the use of more adherent tools to the software development in order to maintain the quality of systems delivered to community. Project management, code versioning, shared components, centralized database and other work processes aim to improve the quality of systems developed. In this paper the tools and processes that helped the organization and the expansion of the process of information systems development in the CPD will be presented.

Keywords: Application development, project management, software quality.

1. INTRODUÇÃO

O Centro de Processamento de Dados da Universidade Federal do Rio Grande do Sul possui 45 anos de história dentro da universidade, durante os quais sempre procurou modernizar os processos de trabalho nas atividades de Tecnologia da Informação. Com a abertura de novos concursos para técnicos-administrativos desde 2008, algumas práticas já bem consolidadas no mercado foram trazidas por novos servidores e consequentemente disseminadas dentro do CPD.

Quando o desenvolvimento e a implantação dessas aplicações é colocado em prática, a universidade passa a ter um sistema corporativo integrado, construído a partir de uma base de dados centralizada, sem redundância e compartilhada entre os setores pertinentes. Essa agilidade na produção e disseminação interna das informações representa um fator decisivo para modernizar a gestão acadêmica, ampliar a qualidade da informação e agilizar os processos administrativos.

Este artigo apresenta o trabalho realizado pelo CPD-UFRGS no processo de desenvolvimento de sistemas, que vem continuamente evoluindo com a implantação de novas aplicações e a aplicação de melhorias nos processos já existentes.

2. METODOLOGIA

O fato de o CPD da UFRGS ter um ambiente centralizado de desenvolvimento de aplicações para a universidade permite que os padrões de desenvolvimento sejam mais uniformes entre as equipes, em uma contínua melhor da padronização de desenvolvimento de software, processo já iniciado em 2011 [1]. Existem processos de trabalho já bastante sedimentados entre as equipes de desenvolvimento. Nas seções que seguem serão descritos cada um deles.

2.1. Gerenciamento de Projetos

Uma parte importante do desenvolvimento de software é a gerência dos projetos, que visa documentar as tarefas, organizar os prazos de entrega, definir as equipes envolvidas, entre outras

atividades. Para gerenciar os projetos de desenvolvimento de software, o CPD adotou a ferramenta *Redmine* [2] em 2008, que é uma ferramenta Web para gerenciamento de projetos.

Existem outras ferramentas interessantes para gerenciamento de projeto, como *Basecamp* [3] e *ChiliProject* [4], mas independentemente da escolha, elas são importantes para o melhor gerenciamento das atividades de desenvolvimento de software. As principais funcionalidades do *Redmine* são estimativa de tempo de projeto, alocação de tarefas para diferentes desenvolvedores, documentação do projeto e integração das tarefas com o versionamento do código.

2.2. Versionamento de Código

Quando se aumenta o número de funcionários trabalhando nos sistemas da universidade, torna-se indispensável o controle das alterações nos códigos das aplicações. O versionamento de códigos-fonte fornece ao analista a possibilidade de controlar os responsáveis por cada mudança e segmentar as entregas/módulos em ramos diferentes do projeto, podendo testar separadamente as alterações.

A solução adotada pelo CPD da UFRGS, também em 2008, foi a ferramenta de versionamento *Subversion*/*SVN* [5], com o uso do aplicativo *TortoiseSVN* [6] como interface para o Windows, pois ela é bastante sedimentada no mercado, integrada com aplicativos em vários sistemas operacionais e também ao software de gerenciamento de projetos.

2.3. Framework de Desenvolvimento

Os padrões de desenvolvimento ajudam a manter uma equipe grande de TI alinhada, facilitando a manutenção dos sistemas e mantendo os códigos uniformes, mesmo para desenvolvedores novos. Além disso, os frameworks já trazem funcionalidades mais robustas para as linguagens de programação, agilizando o processo de desenvolvimento de software.

Em relação à linguagem Java, os frameworks de desenvolvimento *Spring* [7] e *Hibernate* [8] já estão consolidados no mercado, com diversas funcionalidades implementadas e padrões definidos. Entretanto, no setor de desenvolvimento da UFRGS foi adotada a linguagem de programação PHP, especialmente devido à baixa curva de aprendizagem da mesma. Para padronizar o desenvolvimento, o *Framework Yii* [9] foi escolhido por diversos aspectos: utiliza padrão de arquitetura MVC [10]; *design pattern ActiveRecord* [11] para abstração do modelo de dados; compatibilidade com SQL Server, que é o SGBD da base de dados institucional; e facilidade de criação de componentes reutilizáveis.

2.4. Componentes Compartilhados

Com a adoção do *Framework Yii* foi possível criar componentes, tanto das camadas de banco de dados, para padronizar o acesso a dados comuns, como de visualização, para padronizar a interface dos sistemas. Com os componentes compartilhados por todas as aplicações, códigos mais frequentes na aplicação foram centralizados nos repositórios de componentes do CPD, podendo facilmente receber contribuição de qualquer desenvolvedor do setor devido ao versionamento de código.

Os componentes de padronização de interface, por exemplo, que constroem o cabeçalho e o menu de todos os sistemas, agora podem ser atualizados em um único local, propagando eventuais melhorias no layout em todas as aplicações, o que gera menos retrabalho e manutenção.

2.5. Aplicação Modelo

Para agilizar o início de um projeto, foi disponibilizado para os desenvolvedores um esqueleto inicial dos arquivos necessários para começar um projeto nos padrões do CPD, utilizando o *Framework Yii*. Esse modelo já contem a chamada dos componentes de uso padrão, as configurações mais comuns dos projetos e exemplos de uso de componentes de interface. Essa aplicação passa pelo controle de versões e é atualizada sempre que é desenvolvida uma melhor forma de se utilizar ou desenvolver as aplicações.

Uma aplicação modelo também auxilia no aprendizado e uso das boas práticas de programação, visto que ela é uma referência de código-fonte para todo o setor de desenvolvimento.

2.6. Banco de Dados Centralizado

A UFRGS já conta com uma base de dados institucional centralizada desde 1996. Essa abordagem garante que qualquer aplicação acesse os dados mais atualizados e reduz a redundância de dados oriundos da duplicação de tabelas semelhantes em diferentes bancos. O Sistema Gerenciador de Banco de Dados é o *Microsoft SQL Server 2005* e há um controle de permissões diferenciado por usuário para aplicações Web e para aplicações cliente-servidor, com alguns projetos de aplicações Web atualmente possuindo um usuário próprio específico.

2.7. Ambientes de Aplicações

A necessidade de haver ambientes distintos para desenvolvimento e produção de aplicações é bastante clara. Com essa condição, modificações incorretas em ambiente de desenvolvimento não afetam as aplicações que estão em ambiente de produção, acessíveis a toda comunidade universitária. Entretanto, verificou-se a necessidade de criação de outros dois ambientes distintos: de teste e de homologação.

Tal abordagem fez-se necessária devido a diversos problemas que ocorriam, como: desatualização de dados do ambiente de desenvolvimento em relação ao ambiente de produção; alteração de dados diretamente na base de dados do ambiente de produção; vazamento de informações a partir do ambiente de desenvolvimento por parte de bolsistas; etc. Para solucionar esses problemas, hoje existem quatro ambientes de aplicações, descritos abaixo:

- Ambiente de desenvolvimento: contém todas as tabelas do banco com dados sensíveis, como nome, e-mail, endereço, CPF, entre outros, adulterados de tal forma que evite que pessoas sejam identificáveis. Os dados são atualizados por demanda e atualmente estão sem atualização por dois anos por não ter sido considerada uma necessidade até o momento. É acessível por todos desenvolvedores.
- Ambiente de teste: contém todas as tabelas do banco igualmente com os dados sensíveis adulterados, porém com dados sendo atualizados diariamente para corresponder com os dados do ambiente de produção. É utilizado para fazer o teste de software e é acessível a qualquer desenvolvedor por demanda.
- Ambiente de homologação: contém todas as tabelas do banco de dados com os dados sem adulteração e copiados diariamente para corresponder com os dados do ambiente de produção. É de acesso restrito a analistas.
- Ambiente de produção: é o ambiente no qual ficam as aplicações disponibilizadas à comunidade universitária. É de uso restrito a alguns analistas e há uma tendência a restringir ainda mais o seu acesso para garantir a integridade dos dados.

2.8. Teste de Software

O teste de software tem por objetivo auxiliar na garantia de qualidade dos sistemas desenvolvidos pelo CPD para a UFRGS. Para isto, são realizadas verificações e validações funcionais e não funcionais nas aplicações desenvolvidas, além de incentivar boas práticas e uso de ferramental auxiliar quando aplicável.

A partir da documentação do sistema, os casos de teste (ou diretivas de teste) são criados e aplicados ao sistema após sua implementação e implantação no ambiente de teste. Dados representativos para a execução dos casos de teste também são definidos e contemplados na execução dos testes. Com isto, são gerados relatórios com o resultado da execução dos casos de teste e, caso sejam encontrados comportamentos inconsistentes, relatórios de incidentes são criados com a descrição detalhada das falhas encontradas. Estes relatórios são enviados para os desenvolvedores e analistas envolvidos no desenvolvimento da aplicação.

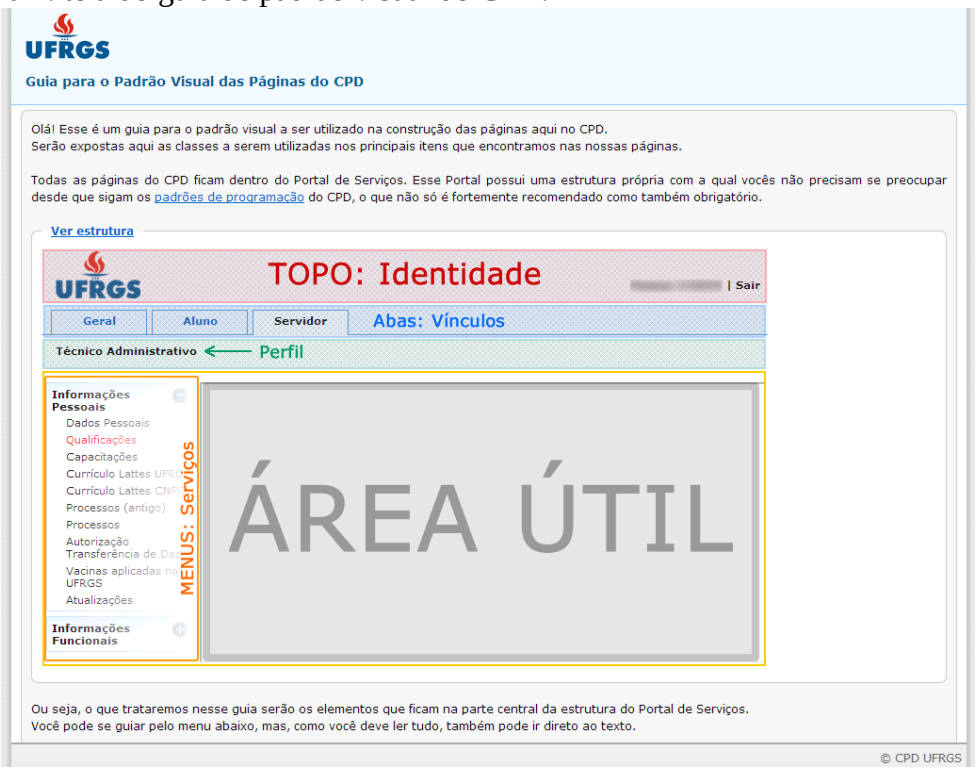
O ciclo de desenvolvimento e teste continua até que os casos de teste sejam satisfeitos e comportamentos inconsistentes não sejam mais encontrados.

2.9. Padrão visual

Os sistemas Web desenvolvidos no CPD até pouco tempo atrás tinham padrões de interface completamente distintos. Isso causava um grande desconforto para o usuário dos sistemas, que se deparavam com interfaces diferentes em um mesmo acesso ao Portal de Serviços da Universidade, tendo até mesmo a impressão de estarem entrando em sistemas diferentes. Para sanar esse problema e modernizar as interfaces, em 2008 criou-se um padrão visual que começou a ser seguido por alguns sistemas. Com a introdução do Framework de desenvolvimento e a forte tendência da padronização do desenvolvimento de software, o padrão visual construído evoluiu e foi instituído oficialmente dentro do CPD, sendo incorporado à aplicação modelo de desenvolvimento, conforme falado previamente.

Para institucionalizar o padrão visual, foi construído um pequeno tutorial identificando e explicando como utilizar as classes criadas. A Figura 1 ilustra a tela de entrada do tutorial. O padrão visual, assim como a aplicação modelo em si, está em constante evolução e, por conseguinte, o tutorial é atualizado na mesma frequência.

Figura 1: tela do guia de padrão visual do CPD.



2.10. Documentação e Treinamento

Junto com o gerenciamento de projeto, o *Redmine* traz uma ótima ferramenta de documentação dos projetos. No projeto de desenvolvimento de componentes compartilhados, é utilizado o recurso de *Wiki* do *Redmine* para documentar como se devem ser utilizados os componentes, melhores práticas de programação e os padrões de desenvolvimento e design do CPD-UFRGS.

Para facilitar o treinamento dos novos bolsistas e servidores, foram desenvolvidas vídeo-aulas [12] sobre as ferramentas e práticas utilizadas no desenvolvimento de software na instituição. Para incentivar a criação das vídeo-aulas, foi realizado um concurso interno para que os funcionários criassem vídeos sobre as ferramentas de trabalho. Há a tendência que o concurso tenha uma reedição.

Além disso, semanalmente os funcionários de desenvolvimento são convidados a participar de uma reunião na qual discutem melhores práticas, novidades no mercado e problemas/soluções enfrentados durante a semana. Essa troca de experiências contribui para o balanceamento e a padronização do desenvolvimento de sistemas.

3. RESULTADOS

Os processos de trabalho descritos permitiram uma melhor qualidade nas aplicações disponibilizadas para a comunidade acadêmica da UFRGS. Quando há a necessidade de corrigir um problema que abrange todos os sistemas, o fato de o desenvolvimento ser centralizado facilita a implementação de uma solução que afete todos os códigos. Outro exemplo é a atualização dos arquivos de layout dos sistemas: por existir um componente centralizado que define a interface, ao modificar esse componente, todos os sistemas são atualizados de acordo com os padrões de design da instituição.

A curva de aprendizado dos bolsistas que trabalham do CPD reduziu drasticamente com a adoção de boas práticas de programação e um padrão de desenvolvimento global na instituição. Assim sendo, também foi reduzido o tempo de desenvolvimento de projetos, bem como os erros de codificação, tanto pelos bolsistas quanto pelo corpo técnico-administrativo desenvolvedor.

As reuniões semanais abriram espaço para uma discussão aberta sobre o desenvolvimento de software e sobre melhores práticas de programação. Esse espaço também é utilizado para solucionar problemas em grupo, facilitando a compreensão da solução e a disseminação do conhecimento para todos os interessados.

4. CONSIDERAÇÕES FINAIS

Mesmo com todas as melhorias implementadas nos últimos anos a busca por qualidade de software nunca tem um fim. Em virtude disso, são pesquisadas novas tecnologias e ferramentas diariamente para garantir as melhoras práticas no desenvolvimento de software.

No processo de melhoria no desenvolvimento de software o CPD está se preparando para utilizar ferramentas para assegurar a qualidade do software como o *Jenkins* [13]. *Jenkins* é uma ferramenta de Integração Continuada para garantir testes unitários, mensuração de qualidade de código, detector de código mal escrito, entre outros recursos.

Além disso, o setor responsável pelo desenvolvimento de software está atualmente mapeando o processo de desenvolvimento. Ao final do mapeamento, espera-se contar com um processo padronizado, otimizado e que tenha seus artefatos de software definidos.

8. REFERÊNCIAS

[1] – MOTTA, T. **Padronização de Metodologias de Desenvolvimento de Software na Divisão de Ensino do CPD da UFRGS.** V Workshop de Tecnologia e Informação das Instituições Federais de Ensino Sperior - WTICIFES, Abril, Florianópolis, 2011.

[2] – Disponível em <http://www.redmine.org/>. Acessado em 02/06/2014.

[3] – Disponível em <https://basecamp.com/>. Acessado em 02/06/2014.

[4] – Disponível em <https://www.chiliproject.org/>. Acessado em 02/06/2014.

[5] – Disponível em <http://subversion.apache.org/>. Acessado em 02/06/2014.

[6] – Disponível em <http://tortoisesvn.tigris.org/>. Acessado em 02/06/2014.

[7] – Disponível em <http://projects.spring.io/spring-framework/>. Acessado em 02/06/2014.

[8] – Disponível em <http://hibernate.org/>. Acessado em 02/06/2014.

[9] – Disponível em <http://www.yiiframework.com/>. Acessado em 02/06/2014.

[10] – Disponível em <http://en.wikipedia.org/wiki/Model–view–controller>. Acessado em 02/06/2014.

[11] – Disponível em <http://en.wikipedia.org/wiki/Active_record_pattern>. Acessado em 02/06/2014.

[12] – Disponível em <http://videos.ufrgs.br/cpd>. Acessado em 02/06/2014.

[13] – Disponível em <http://jenkins-ci.org/>. Acessado em 02/06/2014.