

Desenvolvimento Distribuído de uma Aplicação de Telessaúde com a Metodologia Ágil SCRUM

Alexandre J. H. de O. Luna^{1,2}, Cleyverson P. Costa², Raphael F. A. Patrício², Ana C. M. L. Araújo², Haglay A. N. Silva², João T. F. S. Pessoa², Raony M. Araújo², Marcela B. S. Moraes², Maria S. B. Andrade², Sérgio F. T. O. Mendonça², Fabrício da C. Dias^{1,2}, Marcello R. de Mello^{1,2}, Magdala A. Novaes¹, Jones O. Albuquerque³, Sílvia R. L. Meira²

¹Núcleo de Telessaúde (NUTES), Grupo de Tecnologias da Informação em Saúde, Hospital das Clínicas (HC), Universidade Federal de Pernambuco (UFPE), Brasil.

²Centro de Informática, Universidade Federal de Pernambuco (UFPE), Brasil.

³Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco (UFRPE), Brasil.

Resumo – Este artigo apresenta um estudo de caso completo de Desenvolvimento Distribuído de Software, apoiado na metodologia ágil SCRUM. Este estudo de caso foi realizado através da constituição de uma Fábrica de Software para o desenvolvimento de uma Biblioteca Multimídia, um sistema para o compartilhamento de objetos de aprendizagem utilizados pelos projetos de telemedicina e telessaúde do Núcleo de Telessaúde da UFPE.

Palavras-chave: Fábrica de Software, Engenharia de Software Distribuída, Telessaúde, Telemedicina.

Abstract – This paper presents a complete study case of a Distributed Software Development, supported by the SCRUM agile methodology. This study case was conducted by setting up a Software Factory to develop a Multimedia Library, a system for sharing learning objects used by the telemedicine and telehealth projects of the Center of Telehealth at UFPE.

Key-words: Software Factory, Distributed Software Engineering, Telehealth, Telemedicine.

Introdução

Engenharia de software é uma área do conhecimento voltada para a especificação, desenvolvimento e manutenção de sistemas de software, aplicando tecnologias e práticas da ciência da computação, gerência de projetos e outras disciplinas, objetivando organização, produtividade e qualidade.

Pressman [1] destaca que a Engenharia de software abrange três componentes básicos:

- **Métodos:** proporcionam os detalhes de como construir o software. Englobam tarefas como planejamento e estimativa de projeto, análise de requisitos de software e de sistemas, projeto da estrutura de dados, arquitetura de programa e algoritmo de processamento, codificação, teste e manutenção;
- **Ferramentas:** existem para sustentar cada um dos métodos. Algumas ferramentas existentes para apoio são as Computer-Aided Software Engineering, conhecidas como ferramentas CASE;
- **Procedimentos:** constituem o elo entre métodos e ferramentas. Definem a seqüência em que os métodos são aplicados.

O Desenvolvimento Distribuído de Software (DDS) é um modelo de desenvolvimento de software, que pode ser encarado como uma evolução da Engenharia de Software tradicional e visa o desenvolvimento de produtos de software

através de equipes geograficamente distribuídas. Esta abordagem, em linhas gerais, tem por objetivo diminuir custos, proporcionar a contratação de mão-de-obra qualificada e aumentar a produtividade de desenvolvimento, proporcionando assim um maior poder de competitividade para as organizações [2].

O SCRUM é uma abordagem para desenvolvimento de software, sob o ponto de vista de metodologia ágil. O termo SCRUM vem de um estudo feito por Takeuchi e Nonaka [3]. Como resultado deste estudo, foi percebido que projetos de curta duração, usando equipes pequenas e multidisciplinares (cross-functional), produzem melhores resultados.

O principal objetivo deste trabalho foi a utilização de DDS combinada com SCRUM para o desenvolvimento de uma aplicação de telessaúde, no contexto de Engenharia de Software. Esta abordagem nos pareceu adequada em função da telessaúde ser uma área em plena expansão no Brasil, entretanto, geralmente desenvolvida em instituições de pesquisa e ensino, através de projetos com pequenas equipes de desenvolvimento de software. Outra característica interessante para se experimentar o DDS associado ao SCRUM é o fato da telessaúde ser, em sua essência, uma atividade distribuída, onde as equipes estão separadas geograficamente, e pelo fato de muitas vezes os especialistas no domínio do negócio (saúde) não estarem disponíveis

fisicamente próximos às equipes de desenvolvimento.

Muitas instituições de ensino e pesquisa estão investindo na pesquisa, desenvolvimento da telessaúde, dentre estas se destaca o Grupo de Pesquisa de Tecnologias de Informação em Saúde (TIS) do Núcleo de Telessaúde, sediado no Hospital das Clínicas (NUTES/HC) da Universidade Federal de Pernambuco (UFPE). O NUTES coordena o Projeto da Rede de Núcleos de Telessaúde de Pernambuco (RedeNUTES) [4]. Este projeto está disponibilizando para unidades do programa de saúde da família de municípios de Pernambuco serviços de tele-educação e tele-assistência visando melhorar a resolubilidade destas unidades. Estes serviços permitem a colaboração síncrona e assíncrona entre os profissionais de saúde dos municípios e teleconsultores do Hospital das Clínicas, e produzem um acervo relevante de conteúdos ou objetos de aprendizagem multimídia que podem ser disponibilizados para consulta por outros profissionais de saúde. O projeto Biblioteca Multimídia dataNUTES (BMD), foi concebido pelo NUTES para atender o gerenciamento deste acervo, ampliando ainda mais a capilaridade e benefícios da telessaúde.

Este trabalho utilizou o projeto BMD como estudo de caso para utilização de DDS associado ao SCRUM, e foi desenvolvido pelos alunos da disciplina IN953 – Engenharia de Software do Curso de Mestrado em Ciência da Computação do Centro de Informática da UFPE.

Metodologia

A necessidade apresentada pelo NUTES, denominado Cliente do Projeto, visou o desenvolvimento de uma aplicação que possibilitasse o armazenamento, recuperação, controle e disponibilização inteligente de conteúdo multimídia, denominado Biblioteca Multimídia dataNUTES. O conteúdo desta biblioteca estará disponível nos formatos de áudio, vídeo e texto, o que compreende, entre outros, o gerenciamento de informação obtida através de: sessões de tele-educação, raios-X, tomografias computadorizadas e ultra-sons, visando o aperfeiçoamento de profissionais de saúde [5]. Para esta aplicação, o cliente solicitou, em linhas gerais, servidor de Streaming, Player Web, sistema de busca inteligente, independência de sistema operacional, além da compatibilidade com os sistemas legados (ambiente de ensino e aprendizagem) já implantados.

A estratégia adotada para aplicar DDS associado ao SCRUM, para o desenvolvimento de software, no estudo de caso proposto, foi viabilizada através de uma fábrica de software, denominada *Factory2u*, cuja proposta foi que seus membros trabalhassem em um formato geograficamente distribuído, e tendo seu processo de desenvolvimento de software baseado na

metodologia ágil SCRUM, como apresentado na Figura 1.

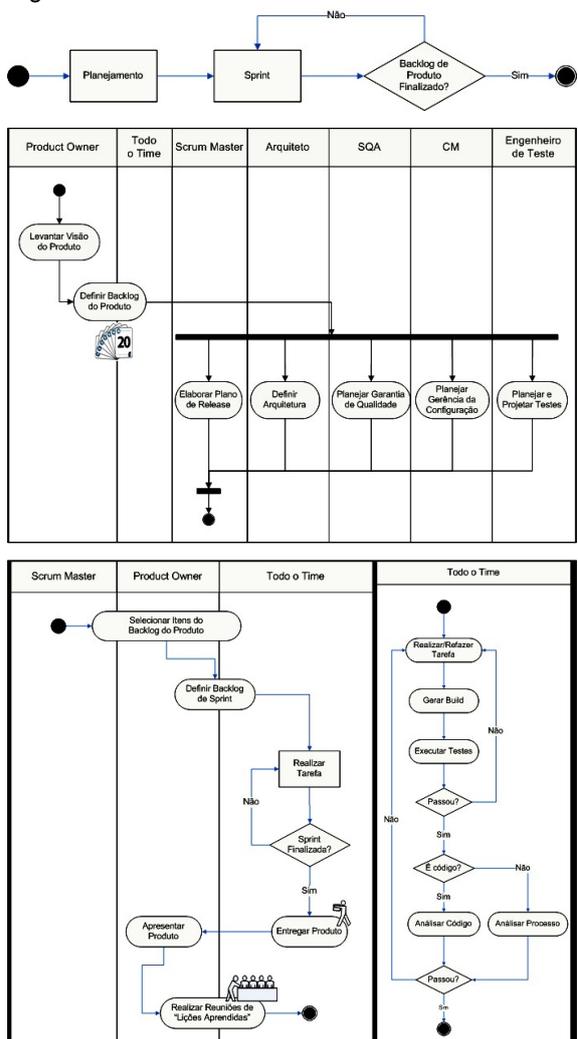


Figura 1: Processo de Desenvolvimento de Software da Fábrica Factory2u

É importante salientar que uma das vantagens das metodologias ágeis em contraposição às metodologias tradicionais é a flexibilidade que estas possuem quando inseridas em ambientes que possuem características como: definição dos requisitos com grande volatilidade (mudanças constantes), onde as equipes são pequenas e os prazos são mais curtos, o que por fim caracteriza a necessidade de um desenvolvimento rápido, sendo este o contexto onde se enquadra o presente trabalho. Outro ponto relevante de ser discorrido é o motivo pela qual foi escolhido SCRUM como metodologia base de desenvolvimento e gerenciamento, uma vez que existem outras metodologias ágeis com características semelhantes. A Extreme Programming (XP) é também uma metodologia ágil para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e que se

modificam rapidamente [7]. Estas duas metodologias, no entanto, se diferenciam pelo foco de atuação. Enquanto o foco do XP é a implementação propriamente dita, ou seja, a codificação, o SCRUM tem seu foco no planejamento e gerenciamento do projeto. Desta forma, considera-se que ambas as metodologias possuem abordagens complementares.

Neste trabalho, apesar de termos utilizado algumas boas práticas do XP nas etapas de implementação do projeto, optamos por ter maior foco nos aspectos de gerenciamento do projeto, que é o ponto forte do SCRUM, onde o principal desafio foi gerenciar a produtividade de uma equipe de nove pessoas, trabalhando fisicamente distribuída, na construção de um produto de software para aplicação em telessaúde.

Inicialmente, os integrantes da fábrica decidiram por estabelecer os dias e horários de reuniões, bem como os meios de comunicação, contendo inclusive, informes de tarefas e relatórios de estado da evolução do projeto, acompanhadas pelas análises comparativas de Velocidade e Execução, através de planilhas e gráficos (Sprint Burndown), conforme demonstrados na Figura 4.

Ao longo execução do projeto fez-se o uso de diversas ferramentas, que viabilizassem o desenvolvimento distribuído, onde, realizamos através de *benchmark*, a escolha pela ferramenta mais indicada à realidade vivida pela Factory2u.

Dentre as ferramentas utilizadas, destacam-se, o Eclipse - Integrated Development Environment (IDE), ambiente de desenvolvimento integrado [8]; Subclipse - plugin para o Eclipse que permite trabalhar com servidores SVN para controle de versão de código e artefatos de software [9]; Cruise Control - framework para o processo de build contínuo [10]; e Ant - ferramenta utilizada para automatizar a construção de testes através de um arquivo no formato XML para descrever o processo de construção e suas dependências [11].

Como linguagem de programação, foi adotado JAVA para o desenvolvimento do componente Cliente/Servidor, responsável pelo armazenamento e conversão dos arquivos multimídia. Para o ambiente responsável pela publicação do conteúdo, foi adotado Python como linguagem de programação, e utilizado o framework Django, devido à sua difundida facilidade de aprendizado e alta produtividade.

Dentre as estratégias de comunicação adotadas, ocorreram poucos encontros presenciais, mas, devido à necessidade de comunicação e incompatibilidade de horários, realizaram-se, com maior frequência, reuniões virtuais (Comunicação Síncrona) com dias e horários pré-definidos (dois encontros semanais), fazendo-se uso de softwares de comunicação VOIP como Skype [12] e Windows Live Messenger [13]. Fez-se também o uso de Lista de Discussão e Emails individuais (Comunicação Assíncrona), de forma constante e sempre que necessário. A quantidade de emails gerados na

Lista de Discussão durante o desenvolvimento do projeto é apresentado na Figura 3.

Outra ferramenta bastante utilizada foi o Google Docs [14], para o acompanhamento das atividades e colaboração online, onde foi criada a planilha de registro de requisitos e atividades, denominada pelo SCRUM, de Ticket Tracking.

Na visão de metodologias ágeis, "estória" é a terminologia empregada para definir "o que" o software deve fazer. Neste contexto uma "estória" atende a um conjunto de características que o software deve obedecer. Fazendo uso das ferramentas apresentadas, as estórias, para desenvolvimento eram geradas pelo cliente, analisadas e aprovadas ou rejeitadas pela Factory2u. Após aprovação, a estória era então encaminhada aos desenvolvedores para implementação, sendo o resultado gerado encaminhado aos testadores para verificação da correção e completude dos requisitos implementados. Após a execução dos testes por parte de integrantes especializados da fábrica, o componente de software era então liberado para teste final por parte do cliente, sendo ele o responsável final pela aceitação. Maiores detalhes sobre o desenvolvimento deste projeto serão abordados na seção de resultados, apresentada a seguir.

Resultados

Existe a necessidade de uma especial atenção na concepção, elaboração e desenvolvimento de uma aplicação web que atenda requisitos de confiabilidade, segurança, e que possibilite controles adequados de acesso, preservando os direitos autorais.

Assim, este trabalho se enquadra nestas características, uma vez que *aborda* o gerenciamento de conteúdo educacional, protegido por direitos autorais, e conteúdos de saúde, protegido pela confidencialidade médico-paciente no caso de conteúdos para teleconsulta ou telediagnóstico. Portanto, o processo de desenvolvimento utilizado, teve papel fundamental para assegurar a qualidade do produto. Neste contexto, serão descritas, a seguir, as seis **Sprints** de execução realizadas no *desenvolvimento* deste projeto. No contexto do SCRUM, cada *sprint* é uma iteração que segue o ciclo PDCA (Planejamento, Desenvolvimento, Controle e Avaliação) e entrega um incremento de software pronto.

A **Sprint 1** se preocupou em desenvolver uma interface gráfica do lado do cliente, para que um arquivo de vídeo pudesse ser enviado para um servidor de Streaming. Além do arquivo propriamente dito, também seriam enviadas informações adicionais sobre o arquivo para serem gravadas no banco de dados do NUTES, sendo priorizado apenas o envio do arquivo e suas informações, sem necessariamente, haver a conversão do vídeo.

Toda a interface foi codificada utilizando-se a biblioteca Swing da J2SE. Não foi encontrado problemas em sua utilização. Um servidor FTP Open Source proporcionou uma significativa ajuda no desenvolvimento das Sprints seguintes, pois foi possível inserir mais operações necessárias aos requisitos, além do recebimento do arquivo.

O principal objetivo da **Sprint 2** foi a conversão dos arquivos de vídeo que seriam enviados ao servidor de Streaming. Como formatos de entrada, a aplicação receberia arquivos em MPEG e AVI. Foi definido, junto ao cliente, que uma resolução de 720 x 480 seria adequada e que num outro momento essa resolução seria parametrizável, uma vez que, para alguns vídeos destinados ao uso em telediagnósticos a qualidade da imagem é um fator crítico. O processo de conversão era iniciado no sistema operacional do cliente, após o término da recepção do arquivo pelo servidor de FTP. No final do projeto, a parametrização da resolução foi abortada por questões de tempo hábil ao cumprimento do projeto.

O cliente se mostrou satisfeito com os produtos das conversões, questionando a estratégia de implementação por entender, num primeiro momento, que o upload e a conversão da mídia tratava-se de uma única operação. Após alguns esclarecimentos e discussões, a estratégia adotada pela factory2u foi aceita, separando em duas operações distintas.

Durante a **Sprint 3**, a equipe definiu qual Player Web de conteúdo multimídia seria utilizado para a reprodução de áudio e vídeo. Foram levadas em consideração portabilidade e facilidade de manutenção, como características desejadas para o Player Web. Finalmente, foi selecionado o Flowplayer, cuja licença é GPL 3, o que se adaptou ao objetivo da equipe e do cliente. Além disso, bugs e pendências das Sprint 1 e 2 também foram tratadas.

Mediante necessidades apontadas na redefinição das histórias, além da dependência e implementação dessas redefinições, algumas histórias da Sprint 4 foram detalhadas já nessa Sprint, facilitando sua posterior execução.

Considerando a **Sprint 4**, num primeiro momento poderia parecer que seriam muitas histórias para uma Sprint de 15 dias, uma vez que seria implementado as histórias de Visualizar Conteúdo, Cadastro/Classificação, Configuração de Acesso e Publicação de Conteúdo.

Como na Sprint anterior, priorizou-se um trabalho de preparação para essa Sprint, referente à história de Visualização de Conteúdo. Em seguida, foi verificado que as histórias Cadastro/Classificação, Configuração de Acesso e Publicação de Conteúdo se complementavam e poderiam compor uma única história, a qual foi denominada unicamente de Cadastro/Classificação.

A partir dessa Sprint, as histórias começaram a ser desenvolvidas em Phython, o que, ao contrário

do que se esperava, dificultou a programação, pois, somente dois integrantes da equipe dominavam a linguagem, e a estratégia para a utilização desta tecnologia, previa, que através da utilização do framework Django, e com uma breve capacitação pelos desenvolvedores mais experientes, os demais entrassem no ritmo planejado de produtividade, que não ocorreu, ficando a codificação do projeto à encargo de menos integrantes que o planejado originalmente. Conseqüentemente, só foi possível concluir a história Visualizar Conteúdo, restando a história de Cadastro/Classificação, que embora iniciado o processo de codificação pelos programadores, não foi concluída, além dos testes da história Visualização de Conteúdo, postergados para a Sprint seguinte.

A **Sprint 5** objetivava concluir todo o projeto, assim, ficou programado para ser finalizada a história de Cadastro/Classificação (unificada com as histórias de Configuração de Acesso e Publicação de Conteúdo) e para ser executada a história de Pesquisa Inteligente.

A falta de conhecimento da linguagem Phython diminuiu a velocidade de programação das histórias. Mesmo assim, a história de Cadastro/Classificação teve seu desenvolvimento concluído. Entretanto, a história Pesquisa Inteligente foi detalhada junto ao cliente, e embora tenha sido iniciado seu desenvolvimento, a finalização além e os testes ficaram pendentes para Sprint seguinte.

A **Sprint 6** (final) foi marcada pelo desenvolvimento da Pesquisa Inteligente, chamada assim por possibilitar a realização de operações com AND, OR, + e -, permitindo pesquisas mais eficientes. Além do desenvolvimento da suíte de testes, base para a validação dos requisitos. Também houve a correção de bugs de Sprints anteriores.

Ainda nesta última Sprint foi programado uma atividade extra – teste de fluxo completo, onde todo o sistema foi novamente testado, a fim de garantir uma maior qualidade e confiabilidade do produto a ser entregue ao cliente.

Após as seis Sprints de desenvolvimento, o cliente recebeu o produto final, dando o aceite como positivo, estando este ainda em fase de implantação. A Figura 2 apresenta algumas interfaces construídas durante o processo de desenvolvimento.

A execução deste projeto proporcionou também a obtenção de informações interessantes a respeito de como foi o processo de comunicação, as mudanças de escopo, velocidade de desenvolvimento e retrabalho.

A Figura 3 apresenta a quantidade de emails (comunicação assíncrona) trocados via lista de discussão durante o projeto.

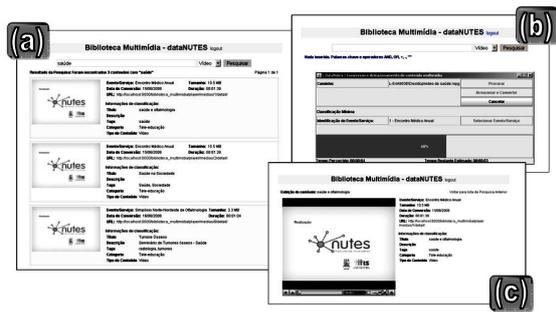


Figura 2: (a) Resultados do sistema de busca por palavras-chave; (b) O módulo de upload e conversão de arquivos multimídia; (c) Exibição do vídeo em modo Streaming, através do Player Web.



Figura 3: Quantidade de emails trocados

Observando o gráfico pode ser identificado que ao longo do período do projeto, de março a julho, a quantidade de troca de emails decresceu. Isto se deveu principalmente em função do amadurecimento da fábrica com relação ao processo de desenvolvimento e entendimento das atividades a serem executadas. O mês de Julho obteve a menor quantidade, pois o projeto já estava em fase de conclusão.

A Figura 4 apresenta a velocidade de desenvolvimento da equipe, com base no planejado versus realizado.

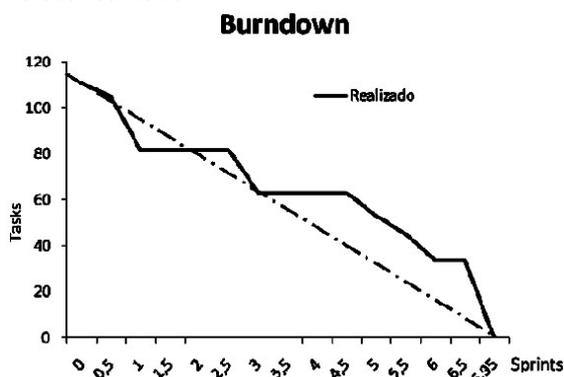


Figura 4: Sprint Burndown em relação às tarefas esperadas, executadas, acumuladas e finalizadas.

Analisando o Sprint Burndown, observa-se que nas primeiras Sprints o executado foi abaixo da média esperada. Esta característica ocorreu, pois o processo de desenvolvimento não estava bem assimilado pela fábrica. Outros fatores identificados

como causadores da baixa produtividade na Sprint 1 foram problemas na comunicação, comprometimento com as reuniões virtuais e mudanças no escopo solicitadas pelo cliente.

Com relação aos testes realizados no sistema, foram gerados 52 casos de teste, compreendidos em 4 suítes de teste, como pode ser visualizado na Figura 5.

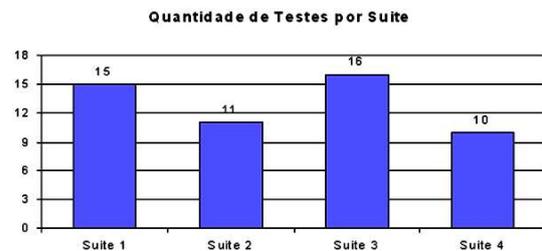


Figura 5: Quantidade de Teste por Suíte

Os testes foram de vital importância para a qualidade final do produto gerado. Outra característica foi a execução dos testes por membros da fábrica que não faziam parte do desenvolvimento. Isto levou a uma independência no processo de testes uma vez que o testador não estava "viciado" com o produto a ser testado.

A quantidade de retrabalho gerado pelo resultado dos testes foi relativamente grande. Da primeira suíte 6 casos de teste falharam, da segunda suíte 4 casos de teste falharam, da terceira suíte 6 casos de teste falharam e da quarta suíte 3 casos de teste falharam. Foi ainda executado uma varredura exploratória onde outras 5 novas falhas foram encontradas. Com base nas suítes de teste, 19 falhas foram encontradas, contabilizando um total de 36,53%, como apresentado na Figura 6. Se adicionado as falhas encontradas pela execução dos testes exploratórios, seria obtido o total de 24 falhas o que corresponde a 42,10% do total de testes realizados.

Quantidade de Testes que Passaram/Falharam (Sem Exploratório)

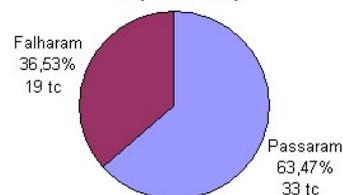


Figura 6: Quantidade de testes que passaram/falharam, sem contar testes exploratórios.

O tamanho final do projeto foi dimensionado em **Linhas de Código** (LOC – Lines of Code), apresentando aproximadamente 8.779 Linhas de Código, representando o desenvolvimento de 128 **story points** determinados pela fábrica através dos Planning Poker e acordadas com o cliente. *Story point* é uma medida relativa do uso de recursos

(dificuldade/ complexidade) que é útil para o planejamento e dimensionamento do tamanho de software.

Discussão e Conclusões

Como pontos positivos da utilização de DDS com SCRUM, podemos destacar a aplicação de um processo simples e objetivo, focado nas necessidades do cliente e não na documentação do projeto, que se demonstrou bastante eficaz. Outro ponto positivo, que podemos destacar, foi o site da fábrica (www.factory2u.net), entre outras ferramentas, se apresentou como mecanismo essencial de comunicação, embora tenha havido, em algumas circunstâncias, certo atraso de sincronização das informações em função do trabalho distribuído da Equipe.

Como ponto negativo, pudemos concluir que se não houver um grau aceitável de compromisso e maturidade profissional na Equipe envolvida no processo, não é recomendável o uso de DDS, pois o processo de comunicação e de iniciativa individual é fundamental no contexto do trabalho de equipes distribuídas.

Como lições aprendidas, podemos destacar o caso da adoção da tecnologia Python, que ao contrário do que se imaginava, não aumentou a produtividade da Equipe, como previsto, pois mesmo com o uso do framework Django, a curva de aprendizado da tecnologia terminou restringindo a implementação da aplicação à poucos integrantes, no final do projeto.

A aplicação gerada neste trabalho foi desenvolvida sobre licença FLOSS - *Free/Libre/Open Source Software*, garantindo que tanto a aplicação, quanto o código-fonte da mesma possam ser utilizados e melhorados pela comunidade de Software Livre Mundial. Sob esta ótica, a produção de uma solução “personalizada” para as necessidades do NUTES/HC-UFPE, foi melhor do que a aquisição de um produto de mercado similar.

Pode-se concluir que a utilização desta metodologia - SCRUM, neste modelo de desenvolvimento - DDS, contribuiu significativamente para a disponibilização rápida de um instrumento de trabalho para unidades que oferecem serviços de telessaúde.

O produto final gerado, Biblioteca Multimídia do dataNUTES, está em fase de implantação no Projeto REDENUTES [6] e estará sendo validada em um ambiente real. Em função desta avaliação poderá vir a ter seu uso ampliado para todos os documentos de imagem e vídeo utilizados pelos projetos do NUTES. Os resultados obtidos com a implementação do módulo de biblioteca digital do sistema dataNUTES poderão vir a ser ampliados com a utilização da biblioteca para gerenciar todos os documentos baseados em imagem e vídeos do NUTES, constituindo-se como um sistema de armazenamento eficiente e de boa qualidade para

as sessões de videoconferência, e ainda permitindo, posteriormente, a elaboração de uma biblioteca de imagens médicas, utilizando a internet como meio para disponibilização.

Agradecimentos

Pesquisa desenvolvida com apoio do Projeto da Rede de Núcleos de Telessaúde de Pernambuco, vinculado ao Programa Telessaúde Brasil, financiado pelo Ministério da Saúde.

Agradecemos a toda Equipe da Factory2u pelo grande desprendimento na realização deste projeto, sem qualquer tipo de remuneração associada, considerando apenas a aprendizagem adquirida no processo como dividendos. Um agradecimento especial ao Raphael F. A. Patrício, sempre presente nos momentos mais difíceis.

Referências

- [1] PRESSMAN, Roger. *Software Engineering: A Practitioner's Approach*, 6ª edição, Mc Graw Hill, 2005.
- [2] Herbsleb, J.D. e Grinter, R. “Splitting the organization and integrating the code: Conway's Law revisited”, In: ICSE 1999, Carolina do Norte, EUA, 1999.
- [3] Takeuchi, H. and I. Nonaka, *The New New Product Development Game*. Harvard Business Review, 1986 (January-February).
- [4] NOVAES, M. A. ; ARAÚJO, K. S.; COUTO, J. M. L. A. A Experiência de Pernambuco em telessaúde. In: Alaneir de Fátima dos Santos; Cláudio de Souza; Humberto José Alves; Simone Ferreira dos Santos. (Org.). *Telessaúde Um Instrumento de Suporte Assistencial e Educação Permanente*. 1 ed. Belo Horizonte: Editora UFMG, 2006, v.1, p. 198-233.
- [5] J. O. A. Segundo, C.H.L.Souza, e M.A.Novoes, “Biblioteca Digital de Videoconferências: Uma Estratégia para Ampliar o Acesso a Programas de Capacitação para o PSF”, CBIS 2006, Florianópolis, 14-18 de outubro de 2006, p.1255-1260.
- [6] Projeto REDENUTES, <http://www.redenutes.ufpe.br>, acessado em 08/08/2008;
- [7] Beck, K. *Programação Extrema Explicada*. Bookman, 1999;
- [8] Eclipse - <http://www.eclipse.org>, acessado em 29/09/2008;
- [9] Subclipse - <http://subclipse.tigris.org>, acessado em 29/09/2008;
- [10] Cruise Control - <http://cruisecontrol.sourceforge.net>, acessado em 29/09/2008;
- [11] Ant - <http://ant.apache.org>, acessado em 29/09/2008;
- [12] Skype - <http://www.skype.com>, acessado em 29/09/2008;
- [13] Windows Live Messenger - <http://get.live.com/messenger>, acessado em 29/09/2008;
- [14] Google Docs - <http://docs.google.com>, acessado em 29/09/2008;

Contato

Alexandre J. H. de O. Luna, Mestrando em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, Pesquisador do Núcleo de Telessaúde da UFPE (NUTES/UFPE), Hospital das Clínicas, Av. Professor Moraes Rego, SN, Cidade Universitária, Recife-PE, (81)2126-3903, Fax: (81) 2126-3904.
E-mail: alexandre.luna@nutes.ufpe.br.