

RELATO DE UM EXPERIMENTO PILOTO DE UMA FÁBRICA DE SOFTWARE BASEADA EM MÉTODOS ÁGEIS

Sérgio Manuel Serra da Cruz¹, Fiamma Quispe², Gustavo Sucupira², Jefferson Leonardo², Lothar Matheus³,
Luis Monsorres², Marcela Yagui², Vanessa Chan², Yara Lima³

Introdução

O conceito *fábrica de software* refere-se a uma coleção estruturada de recursos (humanos, processos e metodologias) relacionados cujo objetivo produzir software ou componentes de acordo com as necessidades específicas do cliente (Pressman, 2011). Uma fábrica de software tradicional (FST) aplica técnicas e princípios para o desenvolvimento de software similares aos de uma unidade fabril qualquer. Ambas se caracterizam como uma linha de montagem de produtos, onde trabalhadores configuram ferramentas, processos e conteúdos usando um modelo com base em um projeto conceitual (Aragon & Souza, 2004).

A adoção indiscriminada de todas as técnicas de uma FST no processo ensino-aprendizagem (Coll & Mira, 1996, Perrenoud, 2000) podem representar um equívoco do ponto de vista da formação de recursos humanos na área de computação. Via de regra, os produtos de software são feitos de forma criativa (individual ou coletivamente) para solucionar um determinado problema ou atender a necessidade de um cliente. Portanto, o nível de padronização tende a ser baixo. Além disso, um segundo problema não menos grave ocorre na FST: posiciona-se o desenvolvedor (muitas vezes alunos de graduação) como parte de uma linha de montagem. Tratar o desenvolvedor (aluno) como um recurso é não só infantilizá-lo como também desumanizá-lo. As FST tratam o processo de trabalho de forma rígida, segmentada e centrada no produto final. O desenvolvedor é apenas um componente do processo fabril. Em resumo, as FST não consideram que o desenvolvedor (aluno) atue como um artífice do conhecimento, nem o consideram como uma mão de obra escassa e extremamente sofisticada que precisa ser motivada para que produza o melhor e que alinhe seus interesses acadêmicos e pessoais com os dos clientes que irão consumir os seus produtos.

Nossa proposta de *Fábrica De Software Baseada Em Métodos Ágeis* (FSMA) se difere do conceito de fábrica de software tradicional (FST). FSMA se alinha com a escola de pensamento proposta por Vygostky (2007) onde a educação é considerada como fonte de desenvolvimento. Aqui se defende que o desenvolvimento ágil de software deva fazer parte do processo de ensino-aprendizado (PEA) focada no aluno, buscando-se a melhoria continuada de forma continuada e iterativa. Portanto, é preciso treinar e manter times de alunos trabalhando juntos, em um mesmo domínio e próximos ao cliente para gerar eficiência e aprendizado na prática, sem abrir mão da qualidade, controlando os custos de desenvolvimento, tempo de entrega do produto.

Este trabalho tem como objetivo apresentar o relato de um experimento piloto de um FSMA centrada no processo ensino-aprendizagem (PEA) e que adotou métodos ágeis como metodologia de desenvolvimento de produtos de software. O experimento-piloto foi conduzido pelos integrantes do grupo PET-SI da UFRRJ durante um intervalo de cinco meses e resultou no desenvolvimento e disponibilização de dois produtos distintos. Os produtos são sistemas Web voltados para atender duas comunidades: (i) alunos do curso de Sistemas de Informação da UFRRJ e (ii) os petianos do PET-SI da UFRRJ.

Material e métodos

A. Métodos Ágeis no Programa PET

A indústria do software é uma das mais importantes nos dias atuais. Emprega milhares de pessoas altamente treinadas e cria alguns dos produtos mais essenciais que nós utilizamos para manter o nosso estilo de vida (Leffingwell & Muirhead, 2004). Neste contexto, o tempo e o custo de desenvolvimento

¹ Tutor do PET-Sistemas de Informação (PET-SI/UFRRJ) da Universidade Federal Rural de Rio de Janeiro

² Graduando do curso de Bacharelato em Sistemas de Informação da Universidade Federal Rural do Rio de Janeiro, membro do grupo PET-SI/UFRRJ.

³ Graduando do curso de Bacharelato em Sistemas de Informação da Universidade Federal Rural do Rio de Janeiro, ex-membro do grupo PET-SI/UFRRJ.

de software passam ser cruciais. Os Métodos Ágeis (MA) de desenvolvimento de software vêm ganhando popularidade desde o início da última década (Melo & Ferreira, 2010). Os MA são regidos pelo “Manifesto Ágil de Desenvolvimento de Software” (MDAS, 2013) e podem ser vistos como uma reação aos métodos tradicionais da Engenharia de Software (conhecidos como métodos dirigidos por planos) que enfatizam o planejamento detalhado e a predição para cada etapa do desenvolvimento. MA se utilizam de equipes ágeis que têm a necessidade de se comunicarem constantemente. Neste sentido, a adoção de MA em uma FSMA se alinha não só com os objetivos gerais do programa PET (estabelecidos na Lei 11.180/2005 e Portarias nº 3.385/2005, nº 1.632/2006 e nº 1.046/2007) como também, representa um excelente instrumento para potencializar o PEA. Especificamente, no que diz respeito ao PET-SI, os MA permitiram: (i) introduzir novas práticas pedagógicas na graduação e (ii) contribuir para a difusão da educação tutorial como prática de formação na graduação. Por fim, os MA possuem ampla abrangência e podem ser facilmente utilizados como ferramenta de apoio no PEA.

B. Programação Extrema como Técnica de Desenvolvimento no PET-SI

Do ponto de vista de MA, é possível adotar duas técnicas distintas de desenvolvimento: *Scrum* (MDAS, 2013) ou programação extrema (XP) (XP, 2013). Neste experimento adotamos apenas a programação extrema (também conhecido como programação em duplas). XP envolve a alocação de dois petianos para trabalhar no desenvolvimento de uma mesma função de um sistema. Eles trabalham em conjunto para apresentar uma melhor solução do problema. Um dos petianos da dupla é responsável por comandar o computador, digitar o código e comandar o par. O outro membro auxilia e revisa o código, seguindo os padrões e especificadas no escopo do projeto do software. As duplas, de tempos em tempos, trocam suas posições. Essa técnica se adapta aos objetivos gerais do programa PET e de uma FSMA. Ela apresenta uma série de vantagens, por exemplo, aumento da comunicação da equipe, geração de código padronizado, manutenção da equipe tecnicamente nivelada, facilidade da entrada de novos petianos, aumento da qualidade do software entregue sem aumentar o prazo de entrega do produto e finalmente, redução do tempo para a produção de documentação do software.

C. Processo Ensino-Aprendizagem em uma FSMA

Historicamente, o processo de ensino-aprendizagem (PEA) tem sido caracterizado de diferentes formas que vão desde a ênfase no papel do professor como difusor de conhecimento, até as concepções atuais que concebem o processo de ensino-aprendizagem com um todo integrado que destaca o papel do aluno (Coll & Mira, 1996, Perrenoud, 2000, Vygotky, 2007). Segundo a ótica da FSMA, o processo de ensino-aprendizagem envolve um *conteúdo que é ao mesmo tempo processo e produto*. Ou seja, parte de um conhecimento formal é oriundo das disciplinas já cursadas pelos alunos ao longo da graduação e outro que é latente, oculto e provém dos esforços de transformação dos conhecimentos pelos alunos em produtos de software que irão atender os seus clientes.

D. Metodologia da FSMA

A metodologia adotada pelo PET-SI abordou conceitos de MA, XP e PEA. Os experimentos foram realizados de modo colaborativo, criando um espaço de referência, onde as funcionalidades dos dois sistemas Web foram estruturadas em função dos saberes previamente adquiridos ao longo do curso de graduação pelos petianos e das necessidades das comunidades de alunos do curso de graduação.

Os experimentos foram executados entre Março de 2013 e Agosto de 2013, contando com a participação de seis petianos, dois ex-petianos, além do tutor. Todos os alunos envolvidos cursavam o último período do ciclo básico do curso de Sistema de Informação da UFRRJ (quarto período). Semanalmente, ocorriam encontros com duração de aproximadamente duas horas com todos componentes. A cada encontro o tutor avaliava (individual e coletivamente): (i) o desempenho dos alunos e suas tarefas de desenvolvimento; (ii) o cronograma dos projetos; (iii) as funcionalidades implementadas em cada sistema Web. No decorrer da semana, os alunos se comunicavam através sistemas *online* (por exemplo, Hangout do Google ou Skype) reportando seus progressos.

Os experimentos desenvolvidos pelos alunos foram compostos pelas seguintes etapas da Engenharia de Software: (i) análise e levantamento de requisitos dos (dois) sistemas; (ii) definição da arquitetura

dos sistemas; (iii) implementação; (iv) testes de funcionalidades e (v) implantação na infra-estrutura de servidores Web administrativos já disponíveis na Universidade.

Resultados e Discussão

Dois produtos foram desenvolvidos na FSMA durante o experimento, a saber: (i) o sistema Web do curso de Sistemas de Informação (<http://cursos.ufrj.br/grad/sistemas/>) (Fig. 1) e (ii) o sistema Web do programa PET-SI (<http://r1.ufrj.br/petsi/>) (Fig. 2). Ambos foram desenvolvidos na plataforma WordPress (2013), um ambiente baseada em software-livre, escrito em PHP e com apoio de banco de dados MySQL, que permite o gerenciamento de conteúdos através da Web.

O primeiro produto tem como objetivo apresentar o curso de Sistema de Informação para a comunidade universitária e também para o público em geral, atuando na atração de futuros novos talentos. Além disso, este sistema possui funções focadas na divulgação de informações sobre as disciplinas do curso, do corpo docente e suas pesquisas, divulgação de eventos acadêmicos, oportunidades de pesquisa, estágios e iniciação científica. O segundo produto tem como objetivo apresentar o próprio grupo PET-SI para o público. Além disso, ele também visa divulgar e dar transparência de todas as ações que estão sendo desenvolvidas no grupo.

Concluindo, a associação dos MA, programação XP e do PEA na FSMA adotada pelo grupo PET-SI neste experimento foi bastante proveitosa. Apresentou as seguintes vantagens: (i) Diluiu o atual paradigma da prática educativa comum a muitos cursos de Sistemas de Informação que apregoam a massificação da formação de alunos-desenvolvedores com pouca ou nenhuma capacidade de resolução de problemas práticos e reduzido poder crítico-reflexivo; (ii) Apresentou aos petianos, o princípio da indissociabilidade entre ensino, pesquisa e extensão, apresentando de forma prática a articulação entre essas três atividades; (iii) Desenvolveu uma atividade acadêmica, em padrões de qualidade e excelência, voltada para os alunos do curso de Sistemas de Informação, contribuindo para a elevação da qualidade da formação dos petianos estimulando seu espírito crítico e reduzindo as distâncias entre a teoria da computação e a sua prática.

Diferentemente de uma FST, a FSMA se estabeleceu sob princípios de desenvolvimento centrados em PEA, buscou-se: (i) Priorizar o indivíduo (aluno) e suas interações ao invés do foco no desenvolvimento de processos formais e uso intensivo de ferramentas de modelagem; (ii) Desenvolver software executável ao invés de documentação; (iii) Produzir respostas rápidas a mudanças nos requisitos ao invés de seguir planos rígidos. Como trabalhos futuros, ocorrerão refinamentos na FSMA e nos dois produtos. Além disso, novos experimentos serão realizados.

Referências

- Aragon, F. A; Teixeira, D. S., *Fábrica de Software-Implantação e Gestão de Operações*. Atlas, 2004. 304p.
- Coll, c.; Miras, M. *A Representação Mútua Professor/Aluno e Suas Repercussões Sobre o Ensino e a Aprendizagem*. Desenvolvimento psicológico e educação: psicologia da educação, ArtMed. 1996. p.265-280.
- Leffingwell, D., Muirhead, D., *Tactical Management of Agile Development: Achieving Competitive Advantage*. 2004.
- Melo, C. O. e Ferreira, G. R. M. *Adoção de métodos ágeis em uma instituição pública de grande porte – Um estudo de caso*. In: *Workshop Brasileiro de Métodos Ágeis, 2010, Anais...Porto Alegre*. 14p.
- MDAS, *Manifesto para Desenvolvimento Ágil de Software*. <http://agilemanifesto.org/iso/ptbr/manifesto.html>. 30 Ago. 2013.
- Perrenoud, P., *Dez Novas Competências Para Ensinar*. Artes Médicas Sul, 2000. 192p.
- Pressman, R., *Engenharia de Software – Uma Abordagem Profissional*. McGraw-Hill, 2011. 780p.
- Vygostky, L. S. *Formação social da mente*. Martins Fontes, 2ª Edição, 2007.

WordPress, <http://br.wordpress.org/>. 30 Ago. 2013.

XP, Extreme Programming. www.extremeprogramming.org/rules/pair.html. 30 Ago. 2013.



Figura 1. Screenshot da versão 1.0 do sistema Web do curso de Sistema de Informação da UFRRJ (<http://cursos.ufrj.br/grad/sistemas/>). O sistema possui 22 funcionalidades distintas que podem ser acessadas por menus e teclas de atalho. Além disso, possui funcionalidades que permitem o gerenciamento dinâmico de conteúdos, tarefa que facilita a manutenção do conteúdo por parte da coordenação do curso.

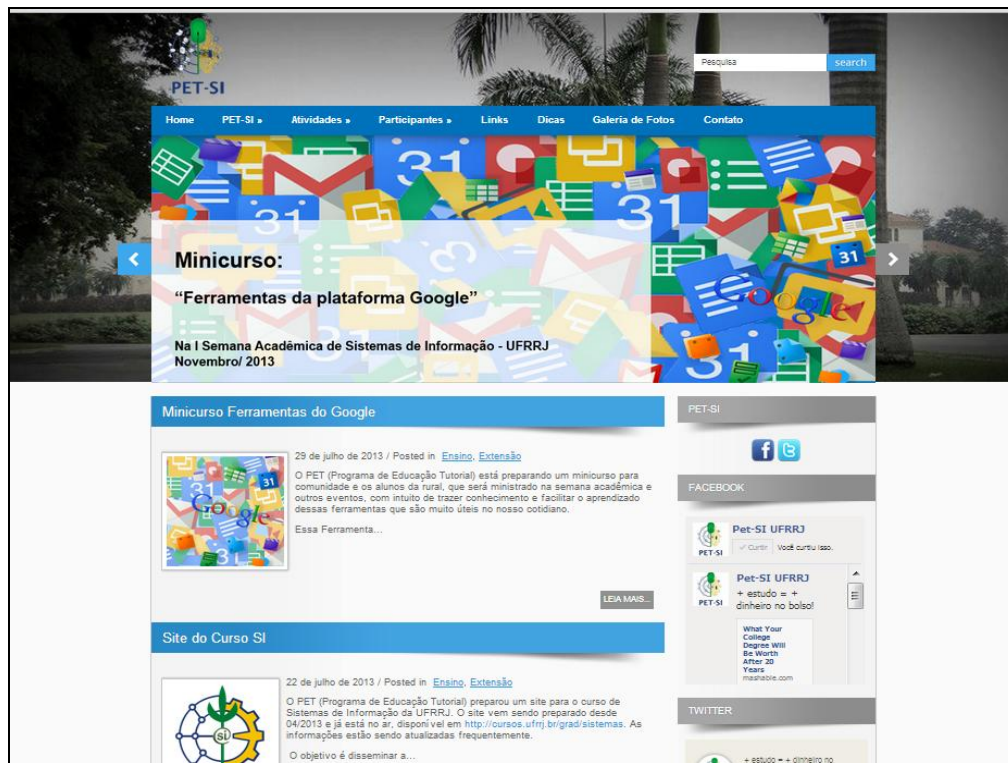


Figura 2. *Screenshot* da versão 1.0 do sistema Web do programa PET-SI (<http://r1.ufrj.br/petsi/>). O sistema possui 15 funcionalidades distintas que podem ser acessadas por menus e teclas de atalho. O sistema está integrado com a página do Facebook do PET-SI (<https://www.facebook.com/PetSIUFRRJ>) e do curso de Sistemas de Informação.