Métodos Ágeis em um Núcleo de Práticas Acadêmico: Relato de Experiência

Camilo Camilo Almendra, Regis Pires Magalhães, Carlos Diego Andrade de Almeida

Campus Quixadá
Universidade Federal do Ceará (UFC) – Quixadá, CE – Brasil

{camilo.almendra,regismagalhaes, diego.andrade}@ufc.br

Abstract. *There is large evidence of agile methods' effectiveness for professional software development. It is of most importance such practices being included in undergraduate Computing curricula. However, agile methods require experience and discipline of teams and its adoption in universities should take into account the natural lack of maturity of novice professionals. This report presents the experience of Center for Informatics Practices in the Federal University of Ceará (UFC-NPI) Quixadá Campus, in the adoption of practices based on agile methods. The practices reported in this work demonstrate that their use in academic environments brings benefits for product quality and students training, although they should be adopted and adapted according to academic needs and constraints.*

Resumo. *Há larga evidência da efetividade do uso de métodos ágeis no desenvolvimento profissional de software, sendo essencial sua incorporação na formação de egressos de cursos da área de Computação. No entanto, métodos ágeis exigem experiência e disciplina das equipes e sua adoção na universidade deve levar em conta a falta de maturidade e de experiência natural aos profissionais iniciantes. Esse relato apresenta a experiência do Núcleo de Práticas em Informática do Campus da UFC em Quixadá na adoção de práticas baseadas em métodos ágeis. As práticas relatadas neste trabalho demonstram que seu uso em ambientes acadêmicos trazem benefícios para a qualidade dos produtos e da formação dos alunos, porém devem ser adotadas e adaptadas de acordo com necessidades e restrições acadêmicas.*

1. Introdução

O efetivo aprendizado de desenvolvimento de software demanda abordagens fortemente permeadas por práticas. A articulação entre teoria e prática é essencial para que iniciantes sejam capazes de compreender a natureza e as inter-relações entre as diversas atividades do desenvolvimento de software [Fox e Patterson 2012]. O estágio profissional promove o contato do aluno com ambientes profissionais ainda no decorrer do curso, contribuindo para solidificar conhecimentos teóricos e práticos aprendidos em atividades curriculares [Cohen e Dann 1995, Gonçalves et al 2013]. A participação em estágio profissional também promove um amadurecimento de habilidades sociais, importantes para profissionais iniciantes de Computação tanto quanto suas habilidades técnicas [Begel e Simon 2008]. Articular teoria e prática de forma colaborativa é

capacidade essencial para egressos de Computação, quer estejam sendo absorvidos por empresas existentes, quer estejam iniciando novos negócios.

A metodologia de trabalho que o estagiário encontra no ambiente profissional pode influenciar na sua formação, embora se espere que um profissional experiente seja capaz de se adaptar a diversos métodos de trabalho. Métodos ágeis promovem uma forma de trabalho que facilita o desenvolvimento de habilidades técnicas e sociais, sendo adotados cada vez mais no mercado brasileiro [Melo et al 2012]. Acompanhando essa tendência, métodos ágeis vêm sendo incorporados aos currículos acadêmicos [Melo et al 2013, Bunse et al 2004].

Essa tendência está presente nos currículos dos cursos de graduação do Campus Quixadá[1], da Universidade Federal do Ceará. O Campus é voltado para cursos em Tecnologia da Informação, e atualmente abriga seis graduações: Sistemas de Informação, Engenharia de Software, Redes de Computadores (tecnólogo), Ciência da Computação, Engenharia de Computação e Design Digital. Nos cursos do Campus, a articulação de teoria e prática é incorporada em todos os currículos por meio de disciplinas com carga horária prática e estágio supervisionado de caráter obrigatório, este realizado em empresas do mercado ou em núcleo de práticas na própria universidade. O Núcleo de Práticas em Informática (NPI) foi estabelecido em 2009, inicialmente chamado de Escritório de Projetos, com o objetivo suprir os setores internos da Universidade com soluções de TI. Posteriormente, o NPI passou também a atender projetos com parceiros externos, tais como bancos públicos e prefeituras. O núcleo se situa no Campus Quixadá da UFC, e vem cumprindo importante papel na articulação entre teoria e prática [Gonçalves et al 2013].

Este artigo apresenta um relato de experiência na adaptação e evolução do processo de desenvolvimento de software do Núcleo de Práticas em Informática. Na seção 2 são discutidos trabalhos relacionados, na seção 3 são discutidos fundamentos para criação e evolução de processos de desenvolvimento em ambientes acadêmicos, na seção 4 é apresentado o processo geral do núcleo, na seção 5 são relatadas práticas ágeis adotadas e como o seu uso influencia na capacitação dos alunos. Por fim, apresentamos nossas considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Na literatura são descritas diversas iniciativas que surgem em universidades com o intuito de melhor preparar o egresso de Computação para o mercado de trabalho. Um ponto comum de muitas dessas iniciativas é a busca por estabelecer situações reais de prática profissional, de forma que a atuação do estagiário seja considerada legítima. Participação legítima em projetos reais complementa a formação adquirida através de trabalhos práticos, visto que por mais elaborados e planejados que sejam esses trabalhos, os mesmos não oferecem a autenticidade da experiência profissional [Begel e Simon 2008]. É crescente a adoção de métodos ágeis na estruturação dos processos de desenvolvimento relacionados a essas iniciativas.

---

[1] Campus da Universidade Federal do Ceará, situado na cidade de Quixadá. Sítio web: http://www.quixada.ufc.br

Métodos ágeis são metodologias de desenvolvimento de software que se baseiam total ou parcialmente nos princípios do Manifesto Ágil[2]. Tais métodos possuem como valores comuns a promoção do planejamento adaptativo, desenvolvimento evolucionário, entrega antecipada, melhoria contínua, e respostas rápidas e flexíveis a mudanças. Na prática, as organizações podem adotar um método por inteiro, ou personalizar seu processo utilizando elementos de vários outros métodos. A adoção de métodos ágeis como arcabouço para ensino de desenvolvimento de software é uma tendência crescente [Goldman et al 2004], no entanto sua incorporação deve levar em consideração que o efetivo uso de métodos ágeis requer disciplina e experiência dos envolvidos [Bunse et al 2004].

Bunse et al (2004) e Goldman et al (2004) apresentam experiências de ensino de programação extrema (XP) em cursos práticos de desenvolvimento de software. Em Bunse et al (2004), alunos foram treinados em métodos tradicionais da engenharia de software e em XP, e posteriormente alocados para desenvolverem um projeto de software. As lições aprendidas apontaram a inerente falta de direcionamento como fator negativo na qualidade do projeto, por conta da inexperiência da equipe. O estudo apontou que o uso de abordagens preditivas pode ser mais efetivo para que os iniciantes adquiram experiência e disciplina, por induzirem um método de trabalho.

Goldman et al (2004) apresenta uma revisão da literatura acerca do ensino de XP na academia, e relata quatro experiências. O estudo discute condições que desfavorecem o uso de XP, além de situações em que sua adoção precisa ser ajustada para ser eficaz. Um dos problemas apontados no ensino de XP é a grande dependência do cliente presente (*on-site customer*) para que as demais práticas funcionem. Um fator motivador identificado foi a escolha do sistema a ser desenvolvido, que deve ser tecnologicamente interessante para os alunos e possuir usuários reais. Diferente do estudo de Bunse, não houve percepção de dificuldade dos alunos em compreender e se engajar no uso das práticas ágeis.

3. Processo de Software em Ambiente Acadêmico

O processo de desenvolvimento de um núcleo de práticas acadêmico deve apresentar características que se adequem ao contexto de estágio supervisionado. Ao realizar estágio, os alunos em sua maioria não apresentam habilidades sólidas de colaboração em equipe, postura profissional, resolução de problemas, pró-atividade, trato com o cliente e habilidades de comunicação. Há naturalmente uma carência de conhecimentos mais práticos em métodos e técnicas de desenvolvimento de software, mesmo que tenham estudado e articulado tais conhecimentos ao longo do curso.

Iniciativas de estabelecimento de núcleos de prática acadêmicos podem ter que lidar com um aspecto importante: o tempo de vínculo do estagiário. O vínculo dos estagiários junto ao núcleo normalmente é curto, já que o mais comum é que os componentes curriculares de estágio estejam presentes nos dois últimos semestres do curso. Isso leva a uma situação peculiar na qual uma grande parte das pessoas envolvidas no desenvolvimento dos projetos passa no máximo um ano na organização.

[2] Manifesto para Desenvolvimento Ágil de Software. Sítio web: http://agilemanifesto.org/iso/ptbr/

A partir de observações durante a execução dos projetos do NPI, percebeu-se que essa rotatividade impacta negativamente a qualidade dos projetos.

A definição e evolução do processo do NPI têm buscado um balanceamento entre três dimensões consideradas prioritárias: gestão do conhecimento organizacional, eficácia do processo e desenvolvimento pessoal dos participantes. A gestão do conhecimento organizacional envolve fomentar a sustentação do ritmo de desenvolvimento dos projetos em face da alta rotatividade das equipes, já que por se tratar de alunos em final de curso esses passam em média seis meses e no máximo um ano atuando no núcleo. A eficácia do processo envolve garantir que os projetos desenvolvidos não sejam tratados de forma semelhante a trabalhos práticos de disciplina, e que efetivamente resultem em produtos e serviços de software úteis aos clientes e patrocinadores. Assim, promove-se o sentimento de realização do estagiário ao ver em funcionamento um produto que ele próprio ajudou a construir. Já o desenvolvimento pessoal visa fornecer um ambiente em que alunos em diferentes níveis de maturidade e experiência possam se desenvolver em ritmos distintos, ao mesmo tempo em que são estimulados a colaborar em equipe. Na iniciativa de criação e evolução do PDS-NPI, essas dimensões exerceram influência na decisão de adotar ou adaptar técnicas comprovadas na indústria.

4. Processo de Desenvolvimento de Software do Núcleo de Práticas em Informática (PDS-NPI)

O processo de software do NPI (PDS-NPI) foi elaborado com base em elementos de vários modelos, tais como Scrum, Programação Extrema e Processo Unificado, e vêm sendo utilizado desde 2011. Uma descrição do PDS-NPI e os resultados alcançados com sua adoção inicial pode ser encontrada em Gonçalves et al (2013). O processo está publicado na Internet e disponível em http://www.npi.quixada.ufc.br/processo.

O PDS-NPI prevê criação e manutenção de artefatos de visão (requisitos de negócios), requisitos funcionais (casos de uso), arquitetura e testes. Essa documentação complementa os artefatos de código e testes automatizados, e visa facilitar o entendimento dos projetos por novos alunos que chegam para trabalhar no núcleo. A gestão de conhecimento baseada em documentos se contrapõe aos princípios do Manifesto Ágil, que sugere "valorizar (...) software em funcionamento mais que documentação abrangente", porém reconhece que há valor na documentação. Entendemos que o próprio código de um sistema é a documentação mais realista e inequívoca sobre o que o sistema faz, e leitura e compreensão de código legado é uma das habilidades fomentadas no núcleo. Entretanto, ter uma visão mais ampla da engenharia de software e ser capaz de entender o valor de artefatos intermediários e seu custo/benefício é também uma habilidade importante a ser incentivada.

Para apoiar a execução dos projetos, o NPI possui uma equipe fixa (com baixa rotatividade) que ajuda a acumular experiências e a repassá-las a novas turmas de estagiários. Assim, aprende-se a partir das experiências das turmas anteriores. No início do estágio, alguns treinamentos e estudos essencialmente práticos são propostos a título de nivelamento dos estagiários com as tecnologias usadas no NPI: Spring Framework, Web (HTML, CSS, JS, jQuery), Git, Maven, Requisitos, Debugging, etc. Nesses treinamentos são apresentados projetos de exemplo com cada tecnologia usada, sempre

partindo de projetos mais simples até alguns mais complexos. Todos esses projetos de exemplo também são disponibilizados publicamente em repositórios do NPI no GitHub. Esses servem de base para o entendimento dos projetos reais que envolvem maior complexidade. Uma Wiki[3] também é disponibilizada com guias, dicas, boas práticas e outros materiais úteis para os estagiários e para a transmissão da base de conhecimento do NPI às suas novas turmas de alunos que se alternam ao longo do tempo, devido a rotatividade dos estágios.

5. Práticas Ágeis

A fim de continuar a evolução do PDS-NPI, uma série de experimentações vem sendo realizadas. Ao longo das atividades do NPI existem mecanismos de coleta de informações que nos ajudam a perceber oportunidades de melhoria: observação do trabalho pelos supervisores, reuniões diárias/semanais e retrospectivas, e seminário final de estágio. Os dois primeiros são mecanismos comuns em equipes ágeis, enquanto que o seminário final é um mecanismo voltado para que cada estagiário faça uma reflexão individual da sua experiência profissional, apontando lições aprendidas e dificuldades encontradas. A análise das informações coletadas nesses momentos leva à identificação de ações de melhoria. A seguir, são apresentadas práticas ágeis adotadas e experimentadas, com uma discussão sobre adaptações necessárias e reflexos gerados no uso do PDS-NPI.

Eventos do Scrum. O conjunto de eventos do Scrum (*sprint*, reunião de planejamento, reunião diária, reunião de revisão e retrospectiva) [Schwaber e Sutherland 2013] foi incorporado ao PDS-NPI desde sua concepção. A escolha do Scrum como arcabouço de base do gerenciamento e execução dos projetos de desenvolvimento foi motivada por três necessidades: manter um conjunto básico de atividades que servisse para vários tipos de projetos; induzir uma filosofia de desenvolvimento iterativa e incremental; e fomentar a participação abrangente dos estagiários em vários aspectos do desenvolvimento do projeto – e não apenas em atividades de implementação de código. As reuniões de revisão com o cliente e as reuniões de retrospectiva são eventos importantes nos quais os alunos recebem *feedback* sobre seu trabalho, e refletem sobre o mesmo, elencando pontos positivos, dificuldades e lições aprendidas. Um sentimento geral normalmente observado é a busca de acertar os passos para fazer melhor a cada nova *sprint*.

Fluxo contínuo. O PDS-NPI é executado normalmente com a configuração de *sprints* de tempo fixo e curto, de cerca de 3 semanas por *sprint*. Essa configuração foi idealizada para promover rápido *feedback* ao time e aos clientes do andamento do projeto. No entanto, em determinados projetos um problema se apresentava recorrente: uma variabilidade grande no tempo necessário para conclusão das tarefas entre os membros das equipes. Essa variação se devia a diferenças no perfil e bagagem técnica de cada estagiário, e levava diretamente a desvios consideráveis entre as estimativas definidas no jogo do planejamento e o esforço real necessário para as tarefas. Esses desvios levaram a necessidade de muitas tarefas inacabadas em uma *sprint* serem constantemente replanejadas para *sprints* posteriores. Enquanto isso não é um problema

[3] https://github.com/npi-ufc-qxd/wiki/wiki

para projetos ágeis em ambientes profissionais, traz um problema pedagógico. Por mais que no estágio o aluno esteja sendo estimulado a buscar resultados através de colaboração e trabalho em equipe, é importante do ponto de vista acadêmico conseguir fazer um supervisão individual dos estagiários. Essa interrupção do trabalho é prejudicial ao desenvolvimento pessoal do estagiário e dificulta a supervisão do seu trabalho. Uma forma de contornar esse problema foi a adoção de fluxo contínuo de trabalho nos projetos onde o problema era mais recorrente.

O fluxo contínuo é um conceito dentro da abordagem Kanban [Kniberg e Skarin 2012] que prega que as atividades da equipe devem fluir ao longo de um fluxo de trabalho, sem se preocupar com prazos limites para conclusão de um grupo de atividades. O conceito de *sprint* de tempo fixo é substituído pelo conceito de marcos de entrega, que são atingidos quando todas as atividades planejadas finalizam. Isso traz a possibilidade de atividades mais demoradas serem finalizadas pelo aluno responsável, sem a interrupção do final da *sprint*. Assim, o aluno completa o ciclo ao seu ritmo, indo do início ao fim de uma atividade que vai gerar um pequeno incremento no projeto. De forma a continuar aproveitando os benefícios das reuniões de revisão e de retrospectiva, ao adotar o fluxo contínuo as reuniões são marcadas regularmente independente do andamento do marco de entrega [Kniberg e Skarin 2012].

Branches e Pull requests. Os projetos desenvolvidos no NPI têm seu código e documentação disponibilizados publicamente sob licença livre em repositórios do NPI no serviço Web de hospedagem de projetos GitHub[4], que tem ganhado força como plataforma de apoio educacional [Zagalsky et al 2015]. Os estagiários diariamente armazenam no repositório remoto as alterações que realizam no código e em documentos dos projetos em que atuam. Também é possível reverter para uma versão anterior de algum arquivo específico ou mesmo de um conjunto de arquivos, caso algum problema tenha sido adicionado em versão mais recente. No início do uso do sistema de controle de versão no NPI, havia uma única versão de cada projeto no repositório remoto. No entanto, eram bastante frequentes os conflitos introduzidos por modificações de diferentes membros do projeto sobre um mesmo arquivo, bem como o envio de código com erros básicos e mesmo com baixa qualidade e legibilidade. Tais problemas levaram à adoção da prática da criação de ramificações (*branches*) específicas para cada funcionalidade desenvolvida por cada estagiário, associada à inspeção e revisão de código através da operação de *pull request*, que passou a fazer parte do *workflow* do projeto. O *pull request* baseia-se na solicitação eletrônica que um membro do projeto faz aos supervisores para que sua funcionalidade seja revisada, aceita e incorporada ao código principal (*branch master*) do projeto. Os membros do projeto são notificados por e-mail de que a solicitação de revisão foi realizada, e um ou mais supervisores do projeto já são antecipadamente designados para realizar as revisões, assim que sejam notificados. Caso o código não esteja com qualidade aceitável, o revisor comenta cada trecho de código com sugestões de como melhorá-lo para que seja aceito. O estagiário é notificado por e-mail sobre os comentários, faz as correções propostas e atualiza o código no repositório remoto. O código, então, é novamente conferido pelo revisor, que pode aprová-lo e incorporá-lo ao código principal do projeto (*branch master*) ou adicionar comentários com as novas melhorias

[4] https://github.com/npi-ufc-qxd/

e correções a serem realizadas. Essa prática de revisão de código mostrou-se extremamente útil no NPI. As revisões levaram os estagiários a uma preocupação maior com a qualidade do seu código antes da submissão de *pull requests*. Os supervisores do NPI passaram a observar os problemas mais frequentes e a indicar boas práticas para solucioná-los.

Propriedade coletiva de código. Essa prática encoraja todos a contribuir com novas ideias em todos os segmentos do sistema em desenvolvimento. Qualquer membro da equipe pode modificar qualquer trecho de código ou configuração para adicionar funcionalidade, corrigir defeitos, melhorar interface ou refatorar [Goldman et al 2004]. Essa prática é importante para fomentar a construção, por cada estagiário, de uma visão completa sobre o sistema. Para tanto, as atividades do projeto são sempre divididas em termos de escopo de funcionalidade, e não em atividades técnicas ou módulos/camadas do sistema. Isso faz com que um estagiário para implementar completamente uma funcionalidade tenha que alterar ou criar artefatos nas várias camadas e módulos do sistema.

Integração contínua. Trata-se de uma prática onde os membros de uma equipe integram seus trabalhos frequentemente. Cada membro deve integrar suas mudanças no mínimo, uma vez ao dia, podendo haver ainda múltiplas integrações. E cada integração seria verificada por um build automatizado para detecção de erros integração mais rapidamente. Muitas equipes acham que esta abordagem conduz a uma significante redução nos problemas de integração, permitindo assim que o time desenvolva software coeso mais rapidamente [Fowler 2006]. É possível obter uma série de benefícios ao se utilizar da prática de integração contínua, tais como redução de riscos, redução de processos manuais repetitivos, permitir melhor visibilidade do projeto, estabelecer uma maior confiança no produto do time de desenvolvimento e coleta de Métricas a cada *build* [Duvall et al 2007]**.** Na busca da obtenção desses benefícios foi implantada uma ferramenta de integração contínua no NPI. Houve um processo de seleção da ferramenta que melhor se enquadraria na realidade do NPI e se obteve grande parte dos benefícios esperados. Dentre eles, o maior destaque ficou para coleta contínua das métricas que foi essencial na melhoria da qualidade do produto. Essas práticas possuem forte ligação com os eventos do Scrum e, principalmente, buscam manter coerente a propriedade coletiva do código.

Programação em pares. Essa prática encoraja desenvolvedores a codificarem e revisarem o código um do outro durante todo o processo de codificação e assim remover problemas dos algoritmos, mantendo a codificação simples e correta. Outra grande vantagem do uso dessa prática se dá em ambientes com códigos muito complexos ou com grande quantidade de regras de negócio, pois se montam pares entre desenvolvedores mais experientes e desenvolvedores menos experientes para se obter uma troca facilitada de experiências entre a equipe e minimizar a necessidade do uso de uma documentação extensa [Kivi et al 2000]. Ao se implantar essa prática foram registrados problemas causados pela falta de diferença de experiência dos participantes, pois os alunos ingressantes ao NPI entravam e saiam do núcleo no mesmo semestre. Houve também dificuldade para fazer uma avaliação individual da participação de cada membro no projeto, por conta dessa prática. Após algumas tentativas frustradas de uso regular da prática, a mesma deixou de ser estimulada. No entanto, os alunos conhecem a

prática a partir de disciplinas práticas dos cursos, e percebeu-se que naturalmente em casos de tarefas mais complexas os estagiários trabalham em par para discutirem e programarem tais tarefas. Esse comportamento trouxe o mesmo benefício esperado na tentativa de adoção regular da prática, e apresentou-se mais adequado por respeitar limitações dos estagiários ao mesmo tempo em que favorece iniciativas de colaboração.

6. Resultados

O NPI provê estágio para alunos dos cursos de graduação do campus, operando em turmas semestrais de estágio, e recebe todo semestre em torno de 30 alunos. Atualmente o núcleo conta com uma analista de sistemas, um técnico em desenvolvimento de software, e dois docentes supervisores, que gerenciam e supervisionam as atividades de estágio. Alunos novatos no NPI participam de um ciclo de nivelamento em processos, ferramentas e tecnologias que dura 3 semanas. Nesse ciclo, os alunos realizam sessões de auto-estudo e treinamentos presenciais com técnicos e supervisores, a fim de conhecer o pacote tecnológico usado nos projetos, as ferramentas de desenvolvimento e de colaboração, o PDS-NPI, assim como tomar contato com os próprios projetos em desenvolvimento. A Figura 1 apresenta uma das salas do núcleo e um registro do seminário de final de estágio.

Figura 1. Sala do NPI (à esquerda) e seminário final de estágio (à direita)

As atividades nas quais os estagiários são envolvidos são todas vinculadas a projetos reais (em contraponto aos trabalhos práticos muito comuns em disciplinas), servindo para atender a demanda da comunidade interna e parceiros, e ao mesmo tempo servindo de prática profissional real. Os projetos do NPI estão relacionados a vários tipos de clientes e áreas de negócio, tais como administração pública, administração acadêmica, saúde, agronomia e atendimento psicopedagógico. Um dos desafios da gestão do NPI é estar sempre captando projetos a fim de manter um portfólio suficiente para atender a demanda de trabalho gerada pela quantidade de alunos participantes. Um ponto importante na seleção de projetos são as características tecnológicas da solução a ser desenvolvida, que deve ser adequada às necessidades acadêmicas de formação do aluno e contribuir para o engajamento dos mesmos nos projetos [Goldman et al 2004].

Conforme observado pelos autores em reuniões de retrospectiva, e no seminário final do semestre, o uso de práticas ágeis é apontado pelos alunos como fator importante para o andamento das atividades e para sua própria formação. Vários deles relatam que mesmo com disciplinas práticas de desenvolvimento e de processo (incluindo métodos ágeis), a experiência de estágio no NPI serve como um consolidador de todos os conhecimentos, levando-os a desenvolver habilidades que ainda não haviam sido exigidas. A maior dificuldade apontada na execução de suas atividades é a falta de

maturidade nas tecnologias utilizadas. Do ponto de vista dos gestores do NPI, isso indica que o processo está cumprindo seu papel de ajudar na colaboração e na organização do desenvolvimento, introduzindo de forma orgânica aspectos essenciais da engenharia de software, tais como gerenciamento de requisitos e de escopo, gerenciamento de testes, e satisfação dos clientes.

Além da formação profissional, alunos saem com um portfólio de projetos que podem ser usados para melhorar seus currículos ou mesmo para serem usados como base para futuros projetos em sua vida profissional. O código fonte e a documentação dos projetos desenvolvidos no NPI são publicados sobre licença livre e, portanto, podem ser livremente estudados, modificados, distribuídos e executados, servindo como base de aprendizado para os estagiários do NPI, bem como para outras pessoas, projetos e atividades. Os alunos também podem adicionar aos seus currículos a participação em tais projetos de software livre com os quais contribuiu durante seu estágio no NPI.

7. Considerações Finais

O estabelecimento de núcleos de prática acadêmicos favorece a complementação da formação dos alunos, de forma a prover um local para prática profissional legítima e favorecer a articulação entre teoria e prática dentro dos currículos de Computação. A escolha do processo e práticas a serem utilizadas deve considerar um balanceamento entre práticas do mercado e questões acadêmicas.

Ao longo da definição e evolução do processo de desenvolvimento do NPI, a eficácia do processo, a oportunidade de desenvolvimento pessoal, e a gestão do conhecimento organizacional têm sido consideradas como base para avaliação da adoção de práticas. Esses aspectos visam, respectivamente, proporcionar experiências legítimas, apoiar a formação dos alunos, e dar sustentabilidade em longo prazo para as atividades do NPI.

As práticas relatadas nesse trabalho demonstram que seu uso em ambientes acadêmicos traz benefícios para a qualidade dos produtos e da formação dos alunos, porém devem ser adotadas e adaptadas de acordo com necessidades e restrições. Além do potencial de melhoria que a prática demonstre em contextos de mercado, deve-se considerar também seu impacto no desenvolvimento pessoal dos alunos e no desenvolvimento organizacional. Nessa perspectiva, percebemos que práticas ágeis exigem experiência e disciplina das equipes e sua adoção deve levar em conta a falta de maturidade e de experiência natural aos profissionais iniciantes. No entanto, por serem práticas de fácil entendimento e exigirem pouco conhecimento prático para iniciar o uso, a abordagem que vêm sendo usada no NPI é da experimentação. Inicia-se com o uso de uma nova prática de forma similar ao uso comum no mercado, e durante o processo são realizadas observações e análise da eficácia da prática.

Como trabalho futuro, pretende-se estabelecer avaliações mais focadas na qualidade dos produtos, a fim de complementar as avaliações qualitativas realizadas através de observação dos supervisores, *feedback* de clientes, retrospectivas com as equipes, e relatos de experiência individuais dos alunos ao término do estágio.

Referências

Cohen, N., Dann, W. (1995) “Using an internal internship to enhance computer science education in a two-year college”. In Proceedings of the 26th SIGCSE technical symposium on Computer science education (SIGCSE '95). ACM, New York, NY, USA, 44-47.

Begel, A., Simon, B. (2008, September). “Novice software developers, all over again”. In: Proceedings of the Fourth international Workshop on Computing Education Research (pp. 3-14). ACM.

Bunse, C., Feldmann, R. L., Dörr, J. (2004) “Agile methods in software engineering education”. In: Extreme Programming and Agile Processes in Software Engineering (pp. 284-293). Springer Berlin Heidelberg.

Duvall, P., Matyas, S., and Glover, A. (2007). “Continuous Integration: Improving Software Quality and Reducing Risk”. A Martin Fowler signature book. Addison-Wesley

Fowler, M. (2006). Continuous integration. http://martinfowler.com/articles/continuousIntegration.html. Acesso em : 25.3.2015.

Fox, A., Patterson, D. (2012) “Crossing the software education chasm”. In: Communications of ACM v. 55, n. 5, pp 44-49.

Goldman, A., Kon, F., Silva, P. J. S., Yoder, J. (2004) “Being extreme in the classroom: Experiences teaching XP”. In: Journal of the Brazilian Computer Society, vol. 10, no. 2, pp. 5-21.

Gonçalves, E.J.T., Bezerra, C.I.M., Almendra, C.C., Sampaio, A.L., Vasconcelos, D.R. (2013), “Núcleo de Práticas em Informática: Contribuindo para a Formação em Sistemas de Informação Através do Desenvolvimento de Projetos de Software”. In: Anais do WEI - XXI Workshop sobre Educação em Computação, Maceió, Brasil.

Kivi, J., Haydon, D., Hayes, J., Schneider, R. and Succi. G. (2000) “Extreme programming: a university team design experience.” In Electrical and Computer Engineering, 2000 Canadian Conference. IEEE

Melo, C.O., Santos, V.A., Corbucci, H., Katayama, E., Goldman, A., Kon, F. (2012). “Métodos ágeis no Brasil: estado da prática em times e organizações”. Relatório Técnico RT-MAC-2012-03. Departamento de Ciência da Computação. IME-USP.

Melo, C.O., Santos, V., Katayama, E., Corbucci, H., Prikladnicki, R., Goldman, A., Kon, F. (2013). “The evolution of agile software development in Brazil”. In: Journal of the Brazilian Computer Society, v. 19, n. 4, pp 523-552.

Schwaber, K., Sutherland, J. (2013), “Um guia definitivo para o Scrum: As regras do jogo”. ScrumGuides.org.

Zagalsky, A., Feliciano, J., Storey, M.A., Zhao, Y. e Wang, W. (2015), “The Emergence of GitHub as a Collaborative Platform for Education”. In: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (CSCW '15).