

Aplicação de eXtremme Programming em ambiente acadêmico: o caso do projeto de formatura

Nathalia S. Patrício, Leandro C. Biazon, Selma S. S. Melnikoff, Roseli de Deus Lopes

Escola Politécnica da Universidade de São Paulo (EPUSP)
São Paulo – SP

{nathalia.patricio, leandro.biazon, selma.melnikoff,
roseli.lopes}@poli.usp.br

Abstract. *This paper is the result of a capstone project. One of the project goals was to test the validity of a set of practical proposals for extreme programming, and experience their consistency when applied in the academic context. We have also documented this experience so that other students who want to work with these methodologies in their projects have an report in which they can base their works, with possible heuristics and adjustments were necessary in this particular case.*

Resumo. *Este artigo é resultado de um projeto de formatura. Uma das metas desse projeto foi testar a validade de um conjunto de práticas propostas pela programação extrema, e experimentar sua consistência quando aplicada no contexto acadêmico. Objetivou-se também documentar essa experiência de forma que outros alunos que queiram trabalhar com essas metodologias em seus projetos tenham um relato no qual se basear, com possíveis heurísticas e adaptações que foram necessárias nesse caso em particular.*

1. Introdução

A adoção de metodologias ágeis em projetos de graduação é ainda incomum em todo o mundo, e no Brasil em particular. Há entretanto diversas iniciativas, e o presente trabalho visa ser uma dessas, apresentando um estudo de caso da adoção da programação extrema em um projeto de conclusão de curso em Engenharia.

Algumas práticas da programação extrema podem ser adotadas nas universidades no processo de desenvolvimento de *software*. Entretanto, um conjunto delas precisa ser adaptada para o contexto acadêmico, pois são voltadas ao desenvolvimento comercial de *software*. Com base nas experiências de outros grupos que adotaram essa metodologia (apresentadas na seção 2), uma série de práticas do XP foram adaptadas no processo de desenvolvimento de um projeto de conclusão de curso em Engenharia e serão apresentadas na seção 3 desse artigo.

2. Programação extrema no contexto acadêmico

Primeiramente, foi feito um levantamento bibliográfico de artigos que descrevessem experiências semelhantes de aplicação de metodologias ágeis na graduação. Foram encontrados alguns relatos dessa natureza, inclusive descrevendo a utilização dessas metodologias em projetos de conclusão de curso.

Schneider e Johnston (2003) avaliam teoricamente a aplicação da programação extrema e a conformidade das práticas com o currículo de Engenharia de Computação

do instituto onde lecionam. Noble et al. (2004) e Keefe e Dick (2004) ministraram disciplinas de projetos de conclusão de curso, nas quais o XP foi apresentado com uma das possíveis metodologias a serem escolhidas pelos alunos.

No artigo de Schneider e Johnston (2003), o XP é descartado como uma metodologia compatível com os objetivos da universidade. Cabe observar que os autores não se baseiam em nenhuma experiência real. Já Noble et al. (2004) e Keefe e Dick (2004) consideram as experiências em seus cursos bem sucedidas, ressaltando a qualidade presente nos projetos desenvolvidos e a melhor interação entre os membros da equipe.

Apesar das relatos positivos envolvendo a aplicação desse método ágil no contexto acadêmico, uma série de adaptações foi necessária para adequá-la aos cursos. As principais dificuldades citadas na utilização dessa metodologia na universidade foram:

- **Área de trabalho compartilhada:** Os laboratórios e salas de aula das universidades não foram projetados para a colaboração entre alunos. Um espaço de trabalho coletivo e informativo não está, em geral, à disposição das equipes.
- **Disponibilidade de tempo:** Na programação extrema, os programadores não devem trabalhar mais que 40 horas semanalmente em um projeto, pois a realização de muitas horas-extras diminui a motivação e disposição para o trabalho. Também devem se envolver com o menor número de projetos simultaneamente, pois perde-se muito tempo e energia na troca de contextos. Essas recomendações são impossíveis de serem seguidas num curso universitário, uma vez que não se pode esperar a dedicação integral de um estudante a apenas uma atividade do curso. Além disso, os horários dos alunos são mais fragmentados do que de um programador profissional.
- **Presença do cliente:** A presença do cliente na equipe de XP, prática que enfrenta resistência mesmo em empresas, é ainda mais difícil na universidade. Duas abordagens costumam ser adotadas para suprir essa necessidade. Uma delas é a realização de projetos para um cliente real, normalmente vinculado à alguma organização não-governamental ou entidade pública, buscando solucionar num projeto de graduação alguma demanda da sociedade. Outra é ter alguém da universidade que cumpra o papel de cliente.
- **Necessidade de treinamento (*coaching*):** Normalmente as equipes não tem experiência anterior com o XP, pois metodologias ágeis não costumam ser abordadas nos cursos de Engenharia e Computação. Isso faz com que a presença do treinador (*coach*) seja importante para a aprendizagem da metodologia.
- **Testes:** As experiências relatam dificuldades na utilização de testes automáticos, seja por problemas culturais (como a pouca importância dada a esse tópico no restante do currículo), seja por aspectos técnicos.
- **Formas de apresentação e avaliação:** Parte das avaliações realizadas nas disciplinas de projeto de conclusão de curso é baseada na documentação levantada por etapas das metodologias tradicionais. Como a criação desses documentos não é parte do processo das metodologias ágeis, outros critérios de avaliação foram necessários nas experiências apresentadas.

3. Relato do processo de desenvolvimento

Durante o projeto de conclusão de curso foi desenvolvido uma rede social para a exposição de projetos de ciências da Feira Brasileira de Ciências e Engenharia (Febrace). A aplicação foi desenvolvida em Python, linguagem que a equipe possuía familiaridade, com uso do framework Django, que não era conhecido. A equipe, composta de dois integrantes, possuía algum conhecimento prévio sobre programação extrema e de algumas de suas práticas que foi adquirido em pequenos projetos durante estágios realizados. O período de desenvolvimento do projeto foi de 1 ano letivo.

Abaixo há uma breve discussão sobre as principais dificuldades citadas na literatura e como, nesse caso de estudo, lidou-se com elas:

- **Área de trabalho compartilhado:** Não foi possível dispor de um espaço de trabalho fixo para a realização das atividades referentes ao projeto. Na tentativa de suprir essa necessidade, foi feito o uso de um conjunto de ferramentas de trabalho colaborativo pela Internet, entre elas um quadro branco virtual e uma ferramenta de acompanhamento de *bugs*. Essas ferramentas supriram a contento as alternativas normalmente adotadas num projeto de XP no gerenciamento de histórias e organização e planejamento da equipe.
- **Disponibilidade de tempo:** Sendo impossível a dedicação exclusiva ao projeto, buscou-se em manter uma carga de trabalho semanal de 8 horas.
- **Presença do cliente:** O papel do cliente foi designado a um pesquisador do laboratório organizador da Febrace, e em conjunto com ele foram escritas os cartões de histórias. O cliente participou também dos *Planning games*, atribuindo prioridade às histórias e definindo o conjunto de cartões a serem implementados numa iteração, e das retrospectivas, nas quais foram apresentadas a funcionalidades desenvolvidas.
- **Necessidade de treinamento (*coaching*):** A metodologia era conhecida pela equipe e, por isso, a necessidade de coaching foi amenizada. Porém, a equipe se limitou a explorar as práticas primárias da programação extrema.
- **Testes:** Foram utilizados testes automáticos, em especial, os testes unitários. Houve uma grande dificuldade na escrita desses testes, principalmente por falta de hábito. A avaliação da equipe é de que esse foi o ponto mais problemático e que não foi completamente superado durante o projeto.
- **Formas de apresentação e avaliação:** Um dos grandes desafios desse projeto foi o uso do XP por parte dos alunos, sem nenhuma modificação na forma de avaliação pelos professores. Sendo assim, foi necessário entregar um documento de especificação de software que, nesse caso, foi substituído por uma listagem de histórias iniciais levantadas com cliente.

Alguns outros aspectos são importantes de serem citados:

- **Ciclos curtos:** O trabalho em ciclos curtos é incentivado em XP, sendo o ciclo de 40 horas o ideal no ambiente corporativo. Como no projeto dedicou-se menos do que 25% desse número de horas semanais, foi necessário encontrar um ciclo de tamanho adequado. Para manter a proporção de horas por ciclo, este deveria possuir quatro semanas no caso estudado. Porém, um ciclo de quatro semanas é muito longo, e ciclos curtos são fundamentais para o acompanhamento correto do projeto. Ciclos curtos evitam atrasos, pois forçam a reflexão constante sobre

o andamento do projeto, e trazem a possibilidade do replanejamento e redefinição de escopo, quando necessário. Um ciclo de quatro semanas teria possibilitado a realização de apenas cinco ciclos durante o projeto, o que permitiria pouco tempo de avaliação e replanejamento. Tendo isso em vista, optou-se por um ciclo de desenvolvimento de três semanas, que apresentou uma relação equilibrada entre tarefas realizadas num ciclo e ciclos no projeto.

- **Retrospectivas:** A retrospectiva é uma reunião periódica na qual é avaliado o período anterior do projeto (entre aquela retrospectiva e a anterior). Nessa reunião há a participação da equipe de desenvolvimento e procura-se levantar coisas que deram certo, coisas que precisam ser melhoradas e idéias (desde do projeto em si até coisas referentes ao ambiente de trabalho) que possam ter surgido. Não há um tempo pré-determinado entre retrospectivas, mas para equipes que trabalham em período integral juntas é aconselhável que ocorram quinzenalmente. Como uma adaptação ao processo, foi decidido que, no caso do projeto em questão, essas retrospectivas ocorreriam mensalmente.
- **Programação pareada:** A dificuldade da prática de programação pareada em projetos acadêmicos se dá por dois motivos. Um é o estranhamento causado por essa prática incomum, que gera resistência de sua adoção pelos programadores. Outra são os horários fragmentados dos alunos, que nem sempre tem agendas coincidentes. Nesse projeto, o primeiro motivo foi rapidamente sanado, pois a resistência ao pareamento é enfraquecida nas primeiras sessões, devido aos benefícios trazidos pela prática, como redução dos erros inseridos no código e a maior rapidez na resolução de tarefas complexas. Já o problema dos horários não pode ser completamente contornado. Das oito horas semanais dedicadas ao projeto, cinco foram dedicadas à programação pareada, e demais se deram por trabalho cooperativo à distância. Também não foi possível o revezamento de pares uma vez que o grupo apresentava apenas dois integrantes.

4. Conclusão

O uso de eXtremme Programming se mostrou aderente ao desenvolvimento de um projeto de formatura, tendo apenas que serem feitas algumas adaptações para o contexto específico, conforme apresentado no artigo. A escolha da metodologia foi decisiva para o sucesso do projeto, tendo em vista o curto tempo para seu desenvolvimento.

Referências

- SCHNEIDER, J.; JOHNSTON, L. extreme programming at universities - an educational perspective. In: Proceedings of the 25th International Conference on Software Engineering. [s.n.], 2003. p. 594–599. Disponível em: <<http://dx.doi.org/10.1109/ICSE.2003.1201242>>.
- NOBLE, J. et al. Less extreme programming. In: ACE '04: Proceedings of the sixth conference on Australasian computing education. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004. p. 217–226. Disponível em: <<http://portal.acm.org/citation.cfm?id=979968.979997>>.
- KEEFE, K.; DICK, M. Using extreme programming in a capstone project. In: ACE '04: Proceedings of the sixth conference on Australasian computing education. Darlinghurst, Australia, Australia: Australian Computer Society, Inc., 2004. p. 151–160. Disponível em: <<http://portal.acm.org/citation.cfm?id=979989>>.