

## Experiência de uso do *eXtreme Programming* (XP) em disciplina de Engenharia de Software

Lis Ângela De Bortoli<sup>1</sup>

<sup>1</sup>Instituto Federal do Rio Grande do Sul (IFRS) – Campus Sertão  
Rodovia RS 135, Km 25 – Distrito Eng. Luiz Englert – 99.170-000 – Sertão – RS –  
Brazil

lis.debortoli@sertao.ifrs.edu.br

**Abstract.** *This article describes an experience of using the agile process eXtreme Programming (XP) with the third level classes, of Software Engineering course in the Systems Analysis and Development program, IFRS, campus Sertão. The dynamic was the development of an application, chosen by the class, using the practices and values of XP. The team was set, the classroom environment was reinvented and students played the roles, along with the teacher. The activity was applied to three classes and the results show that the students could better understand the method, in addition to experience in practice, an actual working environment.*

**Resumo.** *Este artigo descreve uma experiência de uso do processo ágil eXtreme Programming (XP) com turmas do terceiro semestre, da disciplina de Engenharia de Software, do curso de Análise e Desenvolvimento de Sistemas do IFRS, campus Sertão. A dinâmica consistiu no desenvolvimento de uma aplicação, escolhida pela turma utilizando as práticas e os valores do XP. A equipe foi definida, o ambiente da sala de aula foi reinventado e os alunos desempenharam os papéis, junto com a professora. A atividade foi aplicada em três turmas e os resultados demonstram que os alunos conseguiram entender melhor o método, além de vivenciar, na prática, um ambiente real de trabalho.*

### 1. Introdução

Com o intuito de melhorar o desenvolvimento de software, em 2001, foi criada, por um grupo experiente de desenvolvedores, a Aliança Ágil. Os membros desta aliança elaboraram o manifesto ágil (Beck, 2001), um documento com princípios a serem seguidos na criação de processos ágeis.

O primeiro projeto *eXtreme Programming* (XP) iniciou em março de 1996, sendo esta abordagem um dos vários processos ou métodos ágeis, que já provaram ser muito bem-sucedidos na indústria de software, de diferentes tamanhos e em todo o mundo. Criado por Kent Beck, XP é uma evolução da abordagem incremental e enfatiza o desenvolvimento em incrementos, o envolvimento do cliente no processo e a constante melhoria do código (Beck, 2004).

Atualmente, no mundo acadêmico, uma questão que se depara é como preparar estudantes, de forma adequada, para que possam atuar na indústria de software e trabalhar em equipes de forma ágil. Pensando nisso, definiu-se uma dinâmica para ser utilizada nas aulas da disciplina de Engenharia de Software, ministrada no terceiro nível do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas (ADS), do Instituto Federal do Rio Grande do Sul, Campus Sertão. Esta dinâmica baseia-se no

desenvolvimento de uma aplicação utilizando as práticas e valores do método XP, levando para a sala de aula a simulação de um ambiente de trabalho. A dinâmica já foi aplicada com três turmas e cerca de quarenta alunos já participaram da atividade.

A fim de apresentar o trabalho, a seção dois deste artigo aborda o processo ágil XP, seus conceitos, práticas e valores. A seção 3 mostra a ideia da dinâmica, suas características e fases. Na seção 4 os principais resultados e uma breve discussão são explicados. Por fim, demonstra-se as considerações finais e as referências bibliográficas utilizadas.

## 2. O processo ágil eXtreme Programming (XP)

Segundo Teles (2004), o XP é um processo de desenvolvimento ágil que busca assegurar que o cliente receba o máximo de valor a cada dia de trabalho da equipe de desenvolvimento. O processo é baseado em torno de práticas e valores que, em sintonia, asseguram que o cliente receba um software de valor, em curto espaço de tempo. Surgiu, junto com outros processos ágeis, numa tentativa de evitar os problemas frequentes advindos do uso de processos tradicionais ou prescritivos, como atrasos nos cronogramas, extrapolação de custos, insatisfação do cliente e cancelamentos prematuros dos projetos.

O trabalho em equipe é moldado pelos valores fundamentais que são *feedback*, comunicação, simplicidade e coragem. Em conjunto, estes valores enfatizam a necessidade de aprendizado mútuo, compartilhamento, aproximação dos envolvidos, a implementação de apenas aquilo que é suficiente para atender as necessidades e a coragem de acreditar que a utilização das práticas e valores permitem evoluir com segurança e agilidade.

As práticas (Figura 1) são utilizadas para aplicar os valores durante o desenvolvimento de software. Estas práticas devem ser utilizadas em conjunto, pois se complementam, de forma que os pontos fracos de cada uma são superados pelos pontos fortes de outras.



Figura 1. Práticas XP

Fonte: Adaptado de Teles (2004).

Em XP os requisitos são expressos como histórias do usuário que são implementados diretamente como uma série de tarefas, o desenvolvimento é incremental e as entregas são frequentes e em curtos períodos. Mudanças devem ser

encaradas como rotina e não como algo que atrapalha o projeto. O desenvolvimento baseia-se fortemente no trabalho em equipe através de práticas como programação em pares, código coletivo e ritmo sustentável.

A participação do cliente deve ser intensa no projeto, sendo peça fundamental para o sucesso do projeto. Para Pressman (2006), à medida que o trabalho de desenvolvimento prossegue, o cliente pode adicionar histórias, mudar o valor de uma história existente, subdividir histórias e eliminá-las. A equipe XP então reconsidera todas as versões remanescentes e modifica os seus planos adequadamente.

Além do cliente, uma equipe XP é formada por pessoas que exercem papéis: gerente de projetos, *coach*, analista de teste ou testador, redator e desenvolvedor. O gerente de projetos é responsável pelo relacionamento da equipe com o cliente, tem a visão do todo e preocupa-se com aspectos administrativos e burocráticos. O papel do *coach* é assegurar que a equipe respeite e utilize as práticas e valores e o analista de testes é responsável por testar e garantir a qualidade do sistema. A atribuição do redator é manter a documentação do sistema atualizada e o desenvolvedor, por sua vez, analisa, projeta e codifica o sistema, ou seja, é o papel que efetivamente constrói o software.

### **3. A dinâmica na disciplina de engenharia de software**

A engenharia de software é uma disciplina que se preocupa com a aplicação de teoria, conhecimento e prática para o desenvolvimento efetivo e eficiente de sistemas de software que satisfaçam os requisitos dos usuários (ACM/IEEE, 2013). Atualmente, um dos tópicos abordados em disciplinas de engenharia de software de cursos de graduação é métodos ágeis. Questionou-se então, como abordar o assunto de forma prática e que os alunos pudessem entender como sua aplicação pode ser feita em um ambiente diferente da sala de aula, simulando um ambiente de trabalho.

Pensou-se em uma dinâmica que pudesse aliar a teoria e a prática através de um estudo de caso. Sendo assim, iniciou-se com uma explanação sobre o surgimento dos métodos ágeis e sobre os princípios ágeis e após, focou-se no processo XP. A seguir, foi realizada uma pesquisa pelos alunos e uma discussão de aproximadamente dois períodos sobre o processo. Definiu-se então, que os alunos desenvolveriam uma aplicação simples para verificar, na prática o que o XP propõem na teoria. Por serem turmas de um tamanho relativamente pequeno (dez a treze estudantes), a equipe XP foi formada por todos os alunos e pela professora. A atividade foi dividida em três fases principais: planejamento, execução e avaliação.

#### **3.1. Planejamento**

No planejamento a primeira etapa foi um *brainstorming* para escolha da aplicação a ser desenvolvida. A seguir, foram distribuídos os papéis, sendo que para cada aluno (ou mais de um em alguns casos) foi designado um papel. Os papéis de gerente de projetos, desenvolvedor, *coach*, cliente, redator e testador foram distribuídos considerando o perfil dos estudantes, ou seja, procurou-se priorizar as habilidades e competências de cada um. A professora exerceu o papel de cliente e como a disciplina se desenvolve em laboratório de informática, definiu-se que a dinâmica teria que ser em outro local, uma sala de aula, respeitando o que o XP propõem sobre o ambiente de trabalho.

Ainda, na fase de planejamento, foram definidas todas as tecnologias a serem utilizadas no desenvolvimento da aplicação, como hardware, banco de dados e linguagem de programação. As tecnologias foram instaladas previamente em notebooks

e todo o material de apoio foi providenciado pela professora: reserva de sala (com projetor, quadro branco, mesas, cadeiras), canetões, fichas para as histórias, *post-its*, celular (para gravar) e máquina fotográfica. Também foi providenciado, pelos alunos, um lanche para que não precisassem sair da sala durante a fase de execução. Todos os padrões de desenvolvimento foram definidos pela equipe como nomes de tabelas, formulários, variáveis, interface com o usuário, etc.

Nesta fase também foram escritas as histórias do usuário (exemplo na Figura 2) e realizada a modelagem do banco de dados no quadro, que foi posteriormente fotografado pelo redator para integrar a documentação do projeto. Cada estudante comprometeu-se em estudar o seu papel como forma de preparação para a fase de execução.

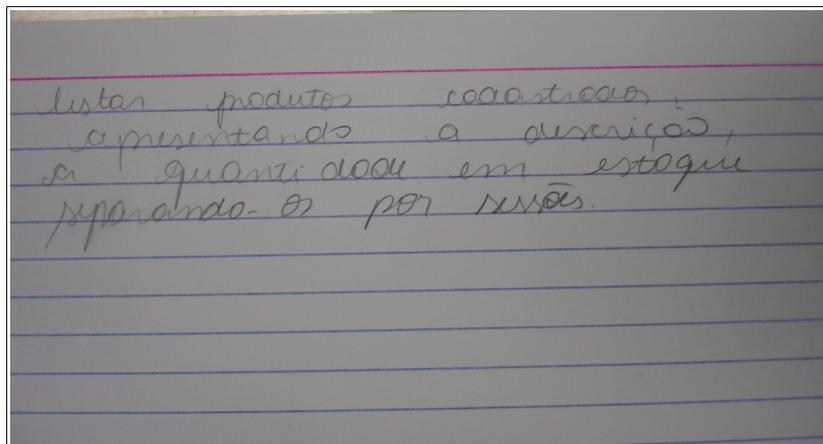


Figura 2. História do usuário

Fonte: Primária.

### 3.2. Execução

A fase de execução consiste na realização da dinâmica na data marcada. Para isso foi montado um cronograma, visto que a disponibilidade de tempo era de um total de quatro horas. O cronograma contempla as seguintes atividades: organização da sala (30 minutos), *stand up meeting* de planejamento e distribuição das histórias (15 minutos), desenvolvimento das histórias (1 hora e 15 minutos), *stand up meeting* de acompanhamento (15 minutos de reunião para relato de andamento e/ou dificuldades e troca de papéis entre os pares), continuação do desenvolvimento das histórias (1 hora e 15 minutos), *stand up meeting* de encerramento (15 minutos), reorganização da sala (15 minutos).

### 3.3. Avaliação

Na fase de avaliação, marcada para uma semana após a execução, os alunos responderam um questionário, cujo objetivo era verificar a percepção deles sobre a dinâmica e sobre o XP. O questionário de avaliação foi dividido em duas partes, sendo que na primeira respondiam sobre a simulação do projeto XP: pontos fortes, dificuldades enfrentadas no geral, o que foi possível aplicar do XP, o que poderia melhorar e atribuição de uma nota para a atividade. A segunda parte abordou o papel que cada um exerceu na simulação do projeto: descrição do papel exercido e o que fez durante a simulação do projeto, de que forma contribuiu com o grupo, dificuldades enfrentadas e autoavaliação. Por fim, os alunos poderiam fazer observações que

julgassem pertinentes.

#### 4. Discussão e resultados

A dinâmica já foi aplicada em três turmas diferentes, entre os anos de 2014 e 2016. A primeira turma desenvolveu um controle para a maratona de engenharia de software, outra dinâmica adotada pela professora. A segunda optou por um controle de um supermercado, abordando os módulos de clientes e vendas. Na terceira turma foi desenvolvido um controle para uma loja de calçados.

O ambiente organizado para a dinâmica foi adequado, a forma de disposição das mesas no centro da sala de forma que todos pudessem se enxergar, facilitou o trabalho dos pares de programação. A utilização do quadro facilitou a interação da equipe, onde foram disponibilizados os modelos elaborados, fixados os padrões e o cronograma, demonstrando-se bastante útil para registrar as decisões dos programadores e as informações que surgiram ao longo do desenvolvimento do trabalho. O quadro de histórias concluídas, em andamento e por fazer, baseado no modelo *Kanban* (Aquino, 2016), também foi montado. Os padrões definidos pelos desenvolvedores deram produtividade ao trabalho. A organização do lanche funcionou, cada um parou conforme sua necessidade. A Figura 3 e a Figura 4 apresentam parte do ambiente de uma das dinâmicas.



Figura 3. O ambiente de trabalho e a equipe

Fonte: Primária.

Com relação aos papéis optou-se por deixar os desenvolvedores como testadores também. O redator também pôde exercer outro papel, pois os registros não necessitam ser realizados a todo momento. Os pares interagiram de acordo com o esperado, mas houve certa resistência no momento da inversão de papéis dos pares. É preciso considerar que no método a troca ocorre após um turno ou dia de trabalho. Os desenvolvedores entenderam adequadamente o sistema proposto pelo cliente, que pôde fazer suas atividades normais e ainda atender as dúvidas dos desenvolvedores. A presença do cliente para esclarecer os requisitos do sistema (histórias) foi essencial e os cartões das histórias foram constantemente atualizados com anotações necessárias.



Figura 4. O ambiente de trabalho e a equipe

Fonte: Primária.

Houve momentos que os desenvolvedores utilizaram o projetor para compartilhar “descobertas” com os demais participantes. A maioria dos pares trabalhou em sintonia e houve interação entre os pares, que se ajudaram nas dúvidas. O gerente de projetos circulou bastante orientando a equipe e o *coach* se preocupou com o andamento dos trabalhos, o cumprimento das práticas e valores. O redator acompanhou o gerente de projetos e registrou, com o detalhe necessário, o andamento do trabalho, mostrando que houve sintonia entre eles. As reuniões em pé foram efetivas e produtivas (Figura 5).



Figura 5. Stand up meeting

Fonte: Primária.

Os valores de *feedback*, comunicação, simplicidade e coragem foram bem marcantes na dinâmica, os estudantes puderam perceber a importância deles para o bom andamento das atividades. As práticas cliente presente, jogo de planejamento, programação em pares, *stand up meeting*, desenvolvimento guiado pelos testes, código coletivo, código padronizado, *desing* simples, metáfora e integração contínua foram todas aplicadas com sucesso. O *refactoring* foi incentivado durante todo o tempo, buscando o melhoramento do código. Já a prática de ritmo sustentável não ficou tão evidente devido ao curto espaço de tempo da dinâmica. Com relação a prática de *releases* curtos puderam verificar que é viável desenvolver, com produtividade, e entregar algo de valor ao cliente num curto espaço de tempo.

A documentação produzida mostrou-se suficiente e adequada como o processo indica. Foi composta pelas histórias do usuário, pelas fotos do quadro, onde estavam o cronograma, os padrões, o quadro das histórias e o modelo de dados. Além disso, todas as reuniões foram gravadas e também foi possível gerar relatórios disponíveis pela linguagem de programação utilizada.

Nas três aplicações da dinâmica foi possível concluir aproximadamente 80% do que tinha sido planejado, ficando os 20% restantes em andamento.

## **5. Trabalhos relacionados**

Alguns trabalhos do uso de métodos ágeis em sala de aula tem sido propostos na literatura. Um deles é proposto por Gomes Filho et al (2014), que aplica conceitos ágeis na disciplina de Métodos Ágeis para os alunos de graduação do curso de Informática. O objetivo é demonstrar como o uso desses conceitos ajuda a criar um ambiente de aprendizagem estimulante e eficaz.

Gomes Filho et al (2015), também desenvolveu um método baseado na simulação de um ambiente de desenvolvimento de sistemas utilizando diversas práticas ágeis em sala de aula, que culmina com num evento anual denominado Sabadágil.

Ainda, o trabalho de Anslow e Maurer (2015) apresenta uma experiência do desenvolvimento de aplicações web, com base em métodos ágeis, ao longo de 13 semanas, para um cliente real, usando a abordagem “aprender fazendo”.

A diferença do trabalho aqui apresentado é que esta é uma prática desenvolvida durante uma ou duas aulas, de forma que os estudantes possam desenvolver parte de uma aplicação, utilizando os conceitos estudados, simulando o ambiente XP. Além disso, os estudantes participam das decisões do projeto: qual aplicação a ser desenvolvida, escolha dos papéis e das tecnologias utilizadas, a modelagem dos dados e gerência de riscos. Desta forma é possível simular um ambiente real de desenvolvimento e os participantes são ativos na dinâmica.

## **6. Considerações finais**

O espaço de sala de aula propicia a formação de saberes e competências. No entanto, é preciso também, preparar os estudantes para o desempenho profissional. Muito tem sido discutido sobre o distanciamento da academia do mundo do trabalho. Por um lado, a necessidade de possibilitar uma sólida formação teórico-prática, de outro às necessidades do mundo profissional. Como conciliar as duas questões, ainda é um desafio. Cursos superiores de tecnologia, são cursos cuja vocação é atender a demandas específicas do mercado de trabalho. Para isso, é preciso adotar metodologias de ensino que procurem contemplar essa importante lacuna existente.

O objetivo deste artigo foi demonstrar, de forma sucinta, uma experiência adotada na disciplina de Engenharia de Software do curso de ADS, utilizando o processo ágil XP.

Após aplicar a dinâmica em três turmas, pode-se concluir que a prática tem-se mostrado efetiva. A receptividade dos alunos é grande, ficam muito espantados com a produtividade atingida e como é importante o trabalho cooperativo. Além disso, é possível proporcionar um diálogo entre os diversos conteúdos abordados no curso, com destaque para banco de dados, linguagens de programação, engenharia de software e gerência de projetos, enfatizando assim a interdisciplinaridade.

A partir do questionário elaborado foi feita uma avaliação da prática. O instrumento contém questões abertas que permitem extrair dados qualitativos. Os estudantes destacaram a rapidez no desenvolvimento, a boa comunicação proporcionada pelo processo, o *feedback* como forma de resolver os problemas no momento que acontecem, a importância da presença do cliente, a possibilidade de ter orientação do *coach* e do gerente de projetos, a produtividade e o comprometimento nas *stand up meetings*, a definição de padrões para se ter um código padronizado, a eficiência do *refactoring*, a adequação da documentação produzida, através de imagens, relatórios e vídeos.

Alguns problemas também aconteceram, como lentidão da rede, ausência de algum estudante, incompatibilidade do par, falta de motivação de alguns membros da equipe, nível de conhecimento diferente da equipe. Estes problemas foram identificados e tratados pelo gerente de projetos de forma a não atrasar o projeto. Assim, os estudantes aprenderam que nem tudo acontece como o planejado, são os riscos que ocorrem em um ambiente de trabalho, que precisam ser identificados, analisados e mitigados.

Os estudantes manifestaram entusiasmo e todos recomendam que esta atividade continue a ser aplicada na disciplina. No entanto, encontra-se em fase de definição um método de avaliação que contemple questões objetivas, de forma a permitir uma avaliação mais precisa da dinâmica.

Como planos futuros pretende-se realizar a dinâmica com outras turmas, buscando o aperfeiçoamento. Também encontra-se em estudo, uma forma de utilizar o processo ágil Scrum nos trabalhos práticos da disciplina.

## Referências

- ACM/IEEE. Software Engineering Curriculum (2013). “Guidelines for Undergraduate Degree Programs in Software Engineering”.
- Anslow, C. Maurer, F. (2015) “An Experience Report at Teaching a Group Based Agile Software Development Project Course”. SIGCSE’15, Kansas City, MO, USA.
- Aquino, R. (2013). “Aplicação de alguns princípios do Kanban em Lean TI”. <http://www.leanti.com.br/artigos/5/aplicacao-de-alguns-principios-do-kanban-em-lean-ti.aspx>. Acesso em Outubro de 2016.
- Beck, K. (2004). Programação Extrema (XP) Explicada. Porto Alegre: Bookman.
- Beck, K. et al. (2001). “Manifesto for Agile Software Development”, <http://agilemanifesto.org/>. Acesso em Fevereiro de 2016.
- Gomes Filho, A. F. et al. (2014). “Usando métodos ágeis para ensinar métodos ágeis.” In: WORKSHOP BRASILEIRO DE MÉTODOS ÁGEIS, 5., 2014, Florianópolis. Anais. São José dos Campos: INPE, p.1–12.
- Gomes Filho, A. F. et al. (2015). “Sabadágil: métodos ágeis de desenvolvimento de software aplicados ao ensino” Revista Brasileira de Informática na Educação, Porto Alegre, v.23, n.3, p.144–163.
- Pressman, R. S. (2006). Engenharia de Software. São Paulo: McGraw-Hill.
- Teles, V. M. (2004). Extreme Programming. São Paulo: Novatec.