

# Aplicando Técnicas de *Grounded Theory* e Retrospectiva Ágil para Buscar Melhorias para o Curso Laboratório XP

Viviane A. Santos<sup>1</sup>, Alfredo Goldman<sup>1</sup>

<sup>1</sup>Instituto de Matemática e Estatística (IME) – Universidade de São Paulo (USP)  
Caixa Postal 66281 – CEP 05315-970 – São Paulo – SP – Brasil

{vsantos, alfredo}@ime.usp.br

**Resumo.** *Este artigo apresenta um estudo empírico sobre o aprendizado de eXtreme Programming em ambiente acadêmico. O estudo emprega duas abordagens distintas para triangulação dos dados: análise qualitativa com grounded theory em um questionário e retrospectiva ágil para investigar o que o curso precisa melhorar na visão dos alunos. A partir dos resultados, ações de melhoria mais direcionadas foram estabelecidas para as próximas edições.*

**Abstract.** *This paper presents an empirical study about learning eXtreme Programming in an academic environment. The study employs two distinct approaches to provide triangulation of the data: qualitative analysis with grounded theory in a questionnaire and agile retrospective to investigate what to improve in the course in the students point of view. From the results, specific improvement actions were established for the next editions of the course.*

## 1. Introdução

Em 2001, os métodos ágeis foram formalmente apresentados pelo *Manifesto Ágil* [Beck et al. 2001] e, desde então, têm recebido muita atenção da indústria e da academia por revelar formas efetivas de desenvolver software.

O movimento pela agilidade no desenvolvimento de software tem provocado uma quantidade substancial de debates e publicações. No entanto, dentro do estado da arte de evidências empíricas, este assunto ainda se mantém escasso [Dybå et al. 2010]. Há poucos estudos com rigor metodológico neste campo e a maioria dos esforços científicos existentes encontra-se no âmbito de *eXtreme Programming* (XP) [Beck and Andres 2004].

[Dybå et al. 2010] relatam que, em ambiente acadêmico, a programação em pares tem promovido redução de 40-50% do tempo estimado para completar uma tarefa, aumento da satisfação no trabalho e qualidade de código. Ao contrário disto, TDD (*Test-Driven Development*) [Beck 2003] tem dispendido maior tempo para a implementação (uma média de 16%), assim como teste antes de codificar (*Test-First Programming*) não acelera a implementação. Segundo eles, a eficiência na programação em pares depende da experiência do programador e da complexidade do sistema e das tarefas envolvidas.

### 1.1. Motivação e Objetivo de Pesquisa

A motivação desta pesquisa consiste no interesse dos responsáveis pelo curso de XP da instituição em melhorá-lo para as próximas edições através de *feedback* dos alunos. A estratégia utilizada para atingir este objetivo considerou duas abordagens distintas:

1. Realização de pesquisa qualitativa através da análise de respostas dos alunos a um questionário não estruturado com questões abertas<sup>1</sup>;
2. Retrospectiva final do curso com a participação de todos os integrantes (professor, *meta-coaches*, *meta-tracker*, *coaches* e desenvolvedores).

Este artigo apresenta um estudo empírico que investiga o que o curso precisa melhorar através da aplicação de técnicas de um método de pesquisa científico em dados de um questionário e da análise do relatório da retrospectiva final. A Seção 2 descreve o curso mencionado. Na Seção 3 é apresentada a primeira abordagem realizada com os alunos do curso, incluindo o uso de técnicas de *grounded theory* para a análise qualitativa. Já na Seção 4 é relatada a segunda abordagem que corresponde a retrospectiva final. A corroboração dos resultados destas abordagens diferentes é a contribuição mais relevante deste artigo. Finalmente, na Seção 5 são apresentadas as ações de melhoria, as conclusões e os trabalhos futuros.

## 2. Laboratório XP: Contexto do Curso

Desde 2001 [Goldman et al. 2004], o curso denominado Laboratório XP tem oferecido aos alunos, de graduação e de pós-graduação em ciência da computação, uma aproximação à realidade de um projeto de desenvolvimento ágil de software.

O curso conta com 1 professor, 3 *meta-coaches* e 51 alunos, sendo 9 *coaches* (dentre eles 1 *meta-tracker*), que correspondem a alunos mais experientes ou que já conhecem a metodologia, e 42 desenvolvedores integrantes de 8 equipes que conduzem projetos reais com clientes reais<sup>2</sup>. Houve um projeto em que atuaram 2 *coaches* e outro em que o *coach* também atuou como *meta-tracker* do curso, auxiliando todas as equipes na condução de *tracking* apropriado às características de seus respectivos projetos. O curso também conta com a dedicação de 8h semanais dos alunos, por este motivo, almoço é oferecido para motivá-los a dispor deste tempo para o projeto. O curso disponibiliza toda a infraestrutura física e tecnológica necessária, e material para a aplicação da metodologia nos projetos.

Na primeira aula foram apresentados o curso, a forma de avaliação e os projetos. Cada aluno escolheu até três projetos desejados para a formação das equipes, além disto, os *coaches* foram escolhidos. Na segunda e terceira aula, XP foi explicada aos alunos. Separadamente, os *coaches* tinham aula com os *meta-coaches* para aprenderem mais sobre como atuar em um projeto ágil. A partir de então, os alunos foram apresentados aos seus respectivos ambientes de trabalho, projetos, *coaches* e clientes.

## 3. Primeira Abordagem: Questionário Aberto

A primeira abordagem corresponde a uma pesquisa qualitativa com base em dados empíricos provenientes de um questionário fornecido aos alunos antes do final do curso a fim de que pudessem expressar com mais fidelidade sua visão sobre o curso. O estudo realizado objetiva entender, pela ótica dos alunos, o que precisa ser melhorado no curso.

### 3.1. Metodologia

A metodologia utilizada nesta abordagem corresponde à coleta de dados por questionário e à análise qualitativa com *Grounded Theory* (GT) [Corbin and Strauss 2007].

---

<sup>1</sup>A forma não-estruturada e indireta do questionário visa incentivar os alunos a projetarem suas opiniões, dificuldades, motivações, crenças, atitudes ou sensações subjacentes sobre os problemas em estudo.

<sup>2</sup>Para detalhes sobre esta edição do curso acesse: <http://www.ime.usp.br/~gold/cursos/2010/xp2010.html>

A coleta foi realizada de forma aberta, não obrigatória e anônima. Os integrantes das equipes foram convidados a responder as seguintes perguntas: O que aprendeu? O que deu certo? O que deu errado? O que poderia melhorar? e entregar suas respostas aos seus respectivos *coaches*. Houve equipes em que cada integrante respondeu o questionário, mas também houve equipes que responderam o questionário em conjunto e outras que não tiveram *feedback* de alguns integrantes, o que resultou em 44 respondentes.

O modo de coleta de dados pode implicar em viés de seleção, devido a possibilidade de omissão dos alunos por se sentirem intimidados em criticar o curso. Porém, todas as equipes forneceram *feedback* representativo, o que minimizou este viés.

Para a análise dos dados qualitativos, GT mais se adequava, devido se fundamentar nos dados, lidar com fenômenos do comportamento humano e permitir extrair aspectos significativos do caso em estudo, ao invés de só transformá-los em números.

### 3.2. Método de Pesquisa Qualitativa *Grounded Theory*

O método de pesquisa qualitativa denominado *Grounded Theory* foi criado por [Glaser and Strauss 1967] com o objetivo de construir teoria indutiva por meio da coleta e análise interativa e sistemática (saturação teórica) de dados empíricos. Diferentemente de outros métodos que usam os dados para avaliar hipóteses predefinidas, este método baseia-se nos dados coletados para gerar uma teoria fundamentada nos mesmos.

De acordo com [Urquhart et al. 2010], este método possui quatro características fundamentais, que são: o seu principal propósito é a construção de teoria; o conhecimento prévio do pesquisador não deve levá-lo à hipóteses pré-formuladas e nem dificultar o surgimento de noções fundamentadas nos dados; a análise e a conceitualização são geradas através de processos intercalados de coleta e análise de dados utilizando a comparação constante; e os códigos são selecionados por um processo de amostragem teórica.

#### 3.2.1. Protocolo

Conforme [Corbin and Strauss 2007], o protocolo deste método consiste em examinar cada documento fornecido, mais de uma vez e independentemente por mais de um pesquisador para evitar viés, buscando identificar unidades de significados para a realização da codificação aberta que, em primeira instância, procura estabelecer **categorias conceituais**. Neste passo, é necessário consenso entre os pesquisadores.

Em segunda instância, são identificados os relacionamentos entre as categorias conceituais para a realização da codificação axial. Nesta etapa, é necessário definir uma lógica de associação entre as categorias. Para tanto, o método de comparação constante é empregado, trabalhando “para frente e para trás” entre as categorias já estabelecidas e os dados originais com o objetivo de identificar fenômenos relacionados, condições causais, contextos, estratégias de ação/interação e consequências das ações para desenvolver **categorias nocionais**. Gráficos podem ser usados para retratar estas relações.

Nesta etapa, também é possível identificar um ou mais fenômenos interessantes (dimensões) dentro da amostragem teórica nos diversos contextos apresentados.

Em terceira instância, é realizada a codificação seletiva para identificar a(s) **categoria(s) principal(is)**, existente ou nova, para desenvolver uma narrativa descritiva (teo-

ria) ao redor dela(s). Em seguida, as interpretações da teoria são validadas e refinadas através de integração teórica, que consiste em relacionar a teoria resultante à outras teorias existentes a fim de gerar confirmações e até formalizações.

### 3.3. Análise dos Dados

Os dados do questionário foram analisados através das etapas de codificação aberta e axial. Para evitar viés, cada pesquisador realizou separadamente a codificação e discordâncias foram discutidas até alcançar consenso.

Como só houve uma coleta de dados, a terceira etapa ainda não foi dada como concluída, devido não ter sido atingida a saturação teórica. Mesmo assim, foi aplicada a técnica de identificação de uma categoria principal para extrair a visão dos alunos sobre o curso nesta edição, o que não significa dizer que foi gerada uma teoria, a priori.

De posse da continuidade desta pesquisa, esta visão dos alunos é de grande utilidade para auxiliar na definição das próximas coletas e análises de dados intercaladas, que, conseqüentemente, culminem na geração de uma teoria fundamentada nos dados.

#### 3.3.1. Codificação Aberta

Nesta etapa foram utilizados códigos “*in vivo*” [Corbin and Strauss 2007], ou seja, baseados na terminologia usada pelos informantes. A codificação aberta resultou em uma matriz de 33 categorias conceituais, com exemplos de unidades de significado dos dados originais e quantidade de ocorrências (#). Por motivos de espaço, a Tabela 1 exibe apenas algumas das categorias conceituais extraídas nesta etapa<sup>3</sup>. O texto sublinhado é uma prévia de relacionamento entre categorias conceituais e é explicado na seção seguinte. Como GT estabelece a necessidade de coletas e análises intercaladas, e neste estudo só houve uma coleta de dados, ainda não foi possível validar as categorias conceituais.

**Table 1. Algumas categorias conceituais selecionadas da codificação aberta**

Id	Categoria Conceitual	Exemplo de Unidade de Significado	#
1	Programação em Pares	<i>“Trabalho em pares facilita aplicação de boas práticas e evita bugs. Além de promover que mais de uma pessoa saiba de <u>todas as partes do código</u>”</i>	24
2	Produtividade	<i>“Aprendemos a programar com mais agilidade (em termos de <u>prática e qualidade</u>)”</i>	10
3	Qualidade de Código	<i>“A meta de 100% de cobertura de <u>testes</u> e familiarização com novas tecnologias (Rails) foram coisas que achei que deram certo, com isso também conseguimos desenvolver novas funcionalidades com código melhor”</i>	6
4	Trabalho em Equipe	<i>“Aprendi a interatuar com o pessoal para trabalhar em <u>colaboração</u>”</i>	53
5	Novas Tecnologias	<i>“O aprendizado de muitas novas tecnologias e ferramentas”</i>	63

#### 3.3.2. Codificação Axial

No início desta etapa, os relacionamentos explicitados pelos alunos foram sublinhados (Tabela 1). Em seguida, utilizou-se um gráfico para ajudar na visualização destes rela-

<sup>3</sup>Para visualizar detalhes da codificação aberta e a lista de categorias conceituais, acesse <http://www.ime.usp.br/~vsantos/pesquisa1/>

cionamentos e na definição da lógica destas ligações para gerar categorias nocionais.

Por motivos de espaço, a Figura 1 apresenta o grafo resultante minimizado com os agrupamentos existentes separados por cores<sup>4</sup>. Primeiramente foram identificados os cliques<sup>5</sup>, depois os ciclos<sup>6</sup> e por último, as ligações simples. Houve categorias, como por exemplo, “Carga Horária do Curso”, que se relacionaram através de ligações simples com diferentes categorias - “Compartilhamento do Conhecimento”, “Novas Tecnologias” e “Teoria sobre XP” - no entanto, os autores escolheram agrupá-la à categoria que possuía maior relevância dentro do contexto, buscando estar sempre fundamentada nos dados. Vários casos de relacionamento não resultaram em agrupamento, devido a força da ligação encontrada nos relatos de determinadas categorias.

Após consenso, foram estabelecidas as categorias nocionais. A Tabela 2 apresenta o identificador da categoria (Id), seu nome juntamente com a cor referente ao seu agrupamento, as categorias conceituais relacionadas, com seus respectivos identificadores à esquerda, seguidas pelo nome e pela quantidade de ocorrências entre parênteses. A Figura 1 mostra o grafo resultante com as cores e o Id referente ao agrupamento das categorias.

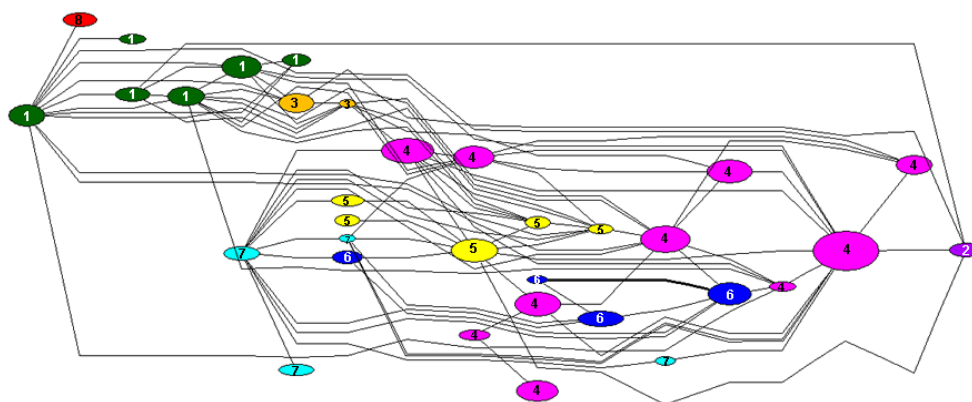


Figure 1. Relação entre as categorias conceituais

Para facilitar o entendimento de cada categoria nocional gerada pelos relacionamentos de categorias conceituais, a narrativa em torno delas é descrita conforme a seguir.

No **Jogo do Planejamento**, o envolvimento do “Cliente” favorece a efetividade da “Reunião de Planejamento”, facilitando o entendimento das “Histórias” e o projeto da solução nas “Reuniões Técnicas da Equipe” na busca de gerar valor incremental em “Iterações Curtas”.

Na categoria **Refatoração e Testes**, os “Testes” influenciam na manutenção do código e propiciam a coragem para realizar “Refatoração”. Na categoria **Fatores Tecnológicos**, o risco da adoção de “Novas Tecnologias” no projeto influenciou na “Produtividade” da equipe, na “Qualidade de Código” e na aplicação de “Padrões de Projeto”, o qual foi atenuado com “Dojo e Superparing”.

Já na categoria **Comunicação e Feedback**, o envolvimento da equipe na

<sup>4</sup>O grafo em tamanho original também está disponível em <http://www.ime.usp.br/~vsantos/pesquisa1/>

<sup>5</sup>Clique é um subgrafo completo que para cada vértice há uma aresta conectando este vértice a cada um dos demais.

<sup>6</sup>Ciclo é um caminho que começa e acaba com o mesmo vértice.

**Table 2. Categorias nocionais resultantes da codificação axial**

<b>Id</b>	<b>Categoria Nocional</b>	<b>Categorias Conceituais Relacionadas</b>
1	Jogo do Planejamento (verde)	9) Cliente (30) 10) Reunião de Planejamento (1) 12) Retrospectivas (11) 13) Histórias (25) 23) Reuniões Técnicas da Equipe (8) 25) Iterações Curtas (15)
2	Boas Práticas de Programação (roxo)	31) Boas Práticas de Programação (5)
3	Refatoração e Testes (laranja)	6) Testes (24) 33) Refatoração (8)
4	Comunicação e <i>Feedback</i> (rosa)	1) Programação em Pares (24) 4) Trabalho em Equipe (53) 8) Ambiente Informativo (8) 11) Comunicação (13) 17) <i>Tracking</i> (27) 18) Horários do Curso (4) 19) Almoço (9) 22) Papo em Pé (3) 26) Estimativas (21) 30) Comprometimento (5)
5	Fatores Tecnológicos (amarelo)	2) Produtividade (10) 3) Qualidade de Código (6) 5) Novas Tecnologias (63) 27) <i>Dojo</i> e <i>Superparing</i> (4) 32) Padrões de Projeto (2)
6	Forma do Curso (azul)	14) Carga Horária do Curso (29) 15) Teoria sobre XP (24) 20) Avaliação (6) 28) Envolvimento do Professor e dos <i>Meta-Coaches</i> (8)
7	Formas de Interação (cyan)	7) Ambiente (9) 21) Reunião de <i>Coaches</i> (5) 24) Compartilhamento de Conhecimento (8) 29) <i>Coach</i> (18)
8	Ambiente de Desenvolvimento (vermelho)	16) Ambiente de Desenvolvimento (10)

“Programação em Pares”, no “Papo em Pé”, na definição de “Estimativas”, no “Almoço”, na atualização do “Ambiente Informativo” e dos gráficos de “*Tracking*”, favorecem a “Comunicação”, o “Comprometimento” e a pontualidade nos “Horários do Curso” dos integrantes para o bom andamento do “Trabalho em Equipe”.

A **Forma do Curso** requer maior “Envolvimento do Professor e dos *Meta-Coaches*” e “Carga Horária do Curso”, mais “Teoria sobre XP” e melhor “Avaliação”. As **Formas de Interação** entre as equipes não deveriam ocorrer apenas na “Reunião de *Coaches*”, devido ao grau de confiança da equipe no “*Coach*”, assim, um melhor uso do “Ambiente” ajuda a intensificar o “Compartilhamento de Conhecimento”.

A categoria **Ambiente de Desenvolvimento** não se encaixou na lógica utilizada para as categorias anteriores, no entanto, os alunos relataram sua ligação com o andamento do projeto, desta forma, utilizando-se da sensibilidade e expertise dos pesquisadores, conclui-se que **Ambiente de Desenvolvimento** inapto afeta o andamento do **Jogo do Planejamento**. Observe que, através da comparação constante, foi identificada uma relação entre duas categorias nocionais.

Da mesma forma, a categoria **Boas Práticas de Programação** também não se ligou fortemente a nenhuma outra categoria a ponto de juntar-se. No entanto, os alunos

relataram que facilitam o desenvolvimento e a manutenção do código-fonte. Portanto, **Boas Práticas de Programação** também favorecem o bom andamento do **Jogo do Planejamento**, influencia na **Comunicação e Feedback**, nos **Fatores Tecnológicos** e na **Refatoração e Testes**.

Após a definição das categorias nocionais, conforme o protocolo, o método de comparação constante é aplicado entre as categorias e os dados originais para estabelecer as dimensões das categorias. A Tabela 3 apresenta as dimensões resultantes.

**Table 3. Dimensões das categorias nocionais**

Id	Categoria Nocional	Dimensões	
		Similaridades	Diferenças
1	Jogo do Planejamento	Eficiência gradativa da equipe	Acompanhamento das deficiências no desenvolvimento
2	Boas Práticas de Programação	Reconhecimento do valor da sua aplicação	Incentivos à sua aplicação
3	Refatoração e Testes	Valorização das práticas	Grau de dificuldade em aplicar
4	Comunicação e Feedback	Eficiência gradativa da equipe	Acompanhamento das deficiências no desenvolvimento
5	Fatores Tecnológicos	Valorização dos fatores tecnológicos	Nível de experiência em tecnologias de mercado
6	Forma do Curso	Necessidade de maior carga horária, mais teoria e maior acompanhamento	Iniciativa dos alunos
7	Formas de Interação	Nivelamento do aprendizado	Acompanhamento das deficiências no desenvolvimento
8	Ambiente de Desenvolvimento	Necessidade de estar inicialmente apto ao desenvolvimento	Experiência em tecnologias de mercado

### 3.3.3. Codificação Seletiva

Esta técnica de codificação foi aplicada para extrair a visão dos alunos sobre o curso nesta edição. Isto não implica em teoria, mas em subsídio para as próximas estratégias de coleta e análise intercaladas, e para identificar o que o curso precisa melhorar.

Assim, as categorias **Jogo do Planejamento**, **Boas Práticas de Programação**, **Refatoração e Testes**, **Comunicação e Feedback**, **Fatores Tecnológicos**, **Forma do Curso**, **Formas de Interação** e **Ambiente de Desenvolvimento** podem ser interpretadas como fatores essenciais ao sucesso do curso. Cada categoria conta parte da história, nenhuma captura-a completamente. Por este motivo, um outro termo ou frase abstrata é necessária, uma idéia conceitual sob a qual todas as categorias são incluídas. Assim, conclui-se que a categoria principal resultante é “Formas de Suporte ao Aprendizado de XP”.

Desta forma, a linha da história para melhorar o curso é a seguinte: *O curso Laboratório XP é de grande valor para os alunos, pois proporciona o aprendizado através do desenvolvimento ágil de um projeto de software em um contexto mais próximo da realidade. No entanto, necessita de mais suporte para um aprendizado mais efetivo de*

*tecnologias, conceitos e práticas; de contínua avaliação dos alunos; e de reconsideração dos pré-requisitos e da carga horária do curso.*

#### **4. Segunda Abordagem: Retrospectiva Final**

Na última aula do curso foi realizada uma retrospectiva ágil adaptada com todos os participantes objetivando coletar dados e refletir sobre o curso para estabelecer uma visão compartilhada de melhorias necessárias para as próximas edições.

##### **4.1. Metodologia**

A dinâmica ocorreu em uma sala de aula com sessões de sete minutos para a discussão civilizada de cada tópico. O professor e os meta-coaches gerenciavam e registravam as sessões em um relatório. Havendo necessidade de estender a discussão, outra sessão de sete minutos poderia ser criada, até atingir consenso.

Esta dinâmica utilizou uma abordagem disciplinada, devido o grande número de participantes e o pouco tempo para discussão (duas horas-aula). Portanto, foram dispostas cinco cadeiras na frente da sala para que interessados em relatar algo sobre o tópico em discussão na sessão pudessem se sentar e expressar sua opinião, em voz alta. As discussões só poderiam ocorrer entre as pessoas sentadas nas cadeiras. O restante da sala deveria ficar em silêncio, sem nenhuma conversa paralela. Dentre as cinco cadeiras, sempre deveria haver uma cadeira livre para o próximo interessado em discutir sobre o tópico. Portanto, se alguém se sentasse na cadeira vazia remanescente, uma das quatro pessoas sentadas, liberaria a sua própria cadeira, visto que já expôs sua visão.

Antes de iniciar as sessões, os alunos sugeriram tópicos para discussão (17 temas). Depois, para cada tema, houve uma votação do número de interessados na sua discussão.

##### **4.2. Resultados**

Por questões de tempo, foram discutidos os cinco temas mais votados na dinâmica. A Tabela 4 lista os temas com o número de votos entre parênteses, algumas transcrições do relatório e o número de sessões realizadas (#). Sempre que terminava uma sessão, as considerações registradas pelo meta-coach eram lidas e confirmadas ou alteradas pelos participantes até atingir consenso. O relatório, gerado a partir da retrospectiva final, descreve fielmente os pontos tratados na dinâmica.

**Table 4. Temas abordados na retrospectiva final**

<b>Id</b>	<b>Tema</b>	<b>Transcrições do Relatório</b>	<b>#</b>
1	Parte teórica (29)	<i>“Espalhar as aulas permitiria ligar teoria à prática”</i>	1
2	Horários da disciplina e dedicação esperada fora dos horários (25)	<i>“Talvez a disciplina merecesse mais crédito e mais aulas por semana.”</i>	1
3	Aprendizado das tecnologias (16)	<i>“Necessidade de muitas tecnologias que não eram dominadas (ex: Eclipse, Ruby, Rails, Git etc.)”</i>	1
4	Participação dos meta-coaches e do professor (12)	<i>“Ter mais proximidade entre o grupo e meta-coaches para ter mais tempo de sentir o que está OK e o que não está”</i>	2
5	Critério de avaliação (10)	<i>“Falta de feedback com critérios subjetivos deixa tudo largado. Muito difícil melhorar.”</i>	2



Os pesquisadores também atuaram na dinâmica como observadores participantes do caso em estudo. Mesmo discutindo apenas cinco temas, muitos temas relacionados também foram mencionados. O que permitiu criar uma noção mais clara das necessidades mais urgentes do curso proveniente do que foi expresso na dinâmica.

Com a análise do relatório, baseada na experiência dos pesquisadores, é possível concluir que, na visão dos alunos, é necessário mais acompanhamento, apoio e *feedback* dos responsáveis pelo curso em relação a aprendizagem de XP, das tecnologias envolvidas e de suas deficiências no desenvolvimento do projeto.

## 5. Conclusão

Dentre as duas abordagens estudadas percebe-se que houve uma interseção entre os assuntos tratados. A combinação das abordagens confere triangulação dos resultados.

De acordo com a observação dos participantes, os projetos do curso apresentaram complexidades variadas, bem como equipes que conseguiram superar os problemas, outras que superaram parcialmente e outras que não os superaram.

Para tanto, como ações de melhoria, são propostas soluções conhecidas, como as dinâmicas de *Coding Dojo* [Sato et al. 2008], retrospectivas e minipalestras com todos os alunos para incentivar a adoção, o aprendizado e o compartilhamento dos valores, princípios e práticas de XP; para prover a exposição mais detalhada do assunto e para detecção/tratamento das deficiências dos alunos durante o desenvolvimento ágil de software; e para facilitar a preparação do ambiente de desenvolvimento.

Outras ações de melhoria são o aumento do número de créditos e, conseqüentemente, da carga-horária do curso; e a reconsideração da forma de composição das equipes, conforme [Sfetsos et al. 2009]<sup>7</sup>, e dos critérios das avaliações periódicas. Além de contar com a teoria da auto-eficácia<sup>8</sup>, visto que [Endres et al. 2009] revela que, em ambientes acadêmicos, o compartilhamento do conhecimento em um grupo depende da interação entre auto-eficácia e conhecimento recíproco esperado de seus integrantes.

Tendo em vista que o sucesso de um projeto ágil depende do tipo de projeto e da equipe, estes devem ser avaliados continuamente durante o curso para proporcionar o suporte necessário às dificuldades enfrentadas.

Este estudo corresponde ao início de uma pesquisa mais abrangente sobre melhoria contínua em métodos ágeis. Como não houve saturação teórica, devido a coleta ter sido feita apenas uma vez e ao final do curso, os pesquisadores pretendem utilizar estes resultados preliminares para especificar melhor as próximas coletas (aprimorar o questionário, a amostragem teórica, utilizar outros tipos de coleta, etc.) e análises intercaladas durante o curso para confirmar ou refutar os resultados. Os autores também consideram construir um repositório de experiências descrito conforme [Meng et al. 2007] e aplicar técnicas de Aprendizagem Organizacional [Argyris and Schon 1978] para identificar formas efetivas de gerar melhorias em organizações ágeis.

---

<sup>7</sup>O autor declara que equipes ágeis mais eficazes são aquelas que consideram a heterogeneidade dos seus integrantes, confirmando a prática XP de time completo (*Whole XP Team*) [Beck and Andres 2004]

<sup>8</sup>A teoria da auto-eficácia explica que o nível de confiança do indivíduo em suas habilidades é um forte motivador para a realização de uma determinada tarefa [Bandura 1997].

## References

- [Argyris and Schon 1978] Argyris, C. and Schon, D. A. (1978). *Organizational Learning: A Theory of Action Perspective*. Addison-Wesley, MA, USA.
- [Bandura 1997] Bandura, A. (1997). *Self-efficacy: The exercise of control*. W. H. Freeman, New York.
- [Beck 2003] Beck, K. (2003). *Test-Driven Development by Example*. Addison Wesley.
- [Beck and Andres 2004] Beck, K. and Andres, C. (2004). *Extreme Programming Explained: Embrace Change*. Addison-Wesley, Boston, 2nd edition.
- [Beck et al. 2001] Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R. C., Mellor, S., Schwaber, K., Sutherland, J., and Thomas, D. (2001). Manifesto for agile software development. Available at: <http://agilemanifesto.org/>.
- [Corbin and Strauss 2007] Corbin, J. and Strauss, A. C. (2007). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. Sage Publications, Inc, 3rd edition.
- [Dybå et al. 2010] Dybå, T., Dingsoyr, T., and Moe, N. B. (2010). *Agile Software Development - Current Research and Future Directions*. Springer, 1st edition.
- [Endres et al. 2009] Endres, M. L., Chowdhury, S., and Rhoad, K. (2009). Effects of student task self-efficacy and expected reciprocity on intent to share knowledge. Paper presented at the Academy of Management Conference. Available at: <http://www.ime.usp.br/~vsantos/pesquisa1/>.
- [Glaser and Strauss 1967] Glaser, B. G. and Strauss, A. L. (1967). *The Discovery of Grounded Theory: Strategies for Qualitative Research*. Aldine Publishing Company, Chicago, IL, USA.
- [Goldman et al. 2004] Goldman, A., Kon, F., Silva, P. J. S., and Yoder, J. (2004). Being extreme in the classroom: Experiences teaching xp. *Journal of the Brazilian Computer Society*, 10:2004.
- [Meng et al. 2007] Meng, X.-x., Wang, Y.-s., Shi, L., and Wang, F.-j. (2007). A process pattern language for agile methods. In *APSEC '07: Proceedings of the 14th Asia-Pacific Software Engineering Conference*, pages 374–381, Washington, DC, USA. IEEE Computer Society.
- [Sato et al. 2008] Sato, D. T., Corbucci, H., and Bravo, M. V. (2008). Coding dojo: An environment for learning and sharing agile practices. *AGILE Conference*, 0:459–464.
- [Sfetsos et al. 2009] Sfetsos, P., Stamelos, I., Angelis, L., and Deligiannis, I. (2009). An experimental investigation of personality types impact on pair effectiveness in pair programming. *Empirical Software Engineering*, 14(2):187–226.
- [Urquhart et al. 2010] Urquhart, C., Lehmann, H., and Myers, M. D. (July 2010). Putting the theory back into grounded theory: guidelines for grounded theory studies in information systems. *Information Systems Journal*, 20:357–381(25).