



UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DEPARTAMENTO ACADÊMICO DE INFORMÁTICA

PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

DISSERTAÇÃO

DISSERTAÇÃO

A INFLUÊNCIA DO DOJO DE PROGRAMAÇÃO NO ENSINO DE PRÁTICAS ÁGEIS

Nome: Ramiro Batista da Luz

Orientador: Adolfo Gustavo Serra Seca Neto

Co-orientador: Robinson Vida Noronha

CURITIBA

2013

RAMIRO BATISTA DA LUZ

**A INFLUÊNCIA DO DOJO DE PROGRAMAÇÃO NO ENSINO DE
PRÁTICAS ÁGEIS**

Orientador: Adolfo Gustavo Serra Seca Neto

Co-orientador: Robinson Vida Noronha

CURITIBA

2013

Dados Internacionais de Catalogação na Publicação

L979 Luz, Ramiro Batista da
A influência do dojo de programação no ensino de práticas ágeis / Ramiro Batista da Luz. — 2013.
67 f. : il. ; 30 cm

Orientador: Adolfo Gustavo Serra Seca Neto.

Co-orientador: Robinson Vida Noronha.

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Computação Aplicada. Curitiba, 2013.

Bibliografia: p. 49-51.

1. Dojo de programação. 2. Programação eXtreme. 3. Programação – Estudo e ensino. 4. Desenvolvimento ágil de software. 5. Tutoria entre pares estudantes. 6. Sistemas de computação interativas. 7. Aprendizagem baseada em problemas. 8. Computação – Dissertação. I. Serra Seca Neto, Adolfo Gustavo, orient. II. Noronha, Robinson Vida, co-orient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Computação Aplicada. IV. Título.

CDD (22. ed.) 621.39

Título da Dissertação

“A INFLUÊNCIA DO DOJO DE PROGRAMAÇÃO NO ENSINO DE PRÁTICAS ÁGEIS”.

por

Ramiro Batista da Luz

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM COMPUTAÇÃO APLICADA - Área de Concentração: Engenharia de Software pelo PPGCA - Programa de Pós-Graduação em Computação Aplicada - Mestrado Profissional – da Universidade Tecnológica Federal do Paraná - UTFPR - Câmpus Curitiba, s 09:00 horas do dia 26 de agosto de 2013. O trabalho foi aprovado pela Banca Examinadora, composta pelos professores:

PPGCA

**Programa de Pós-Graduação
em Computação Aplicada**

Prof. **Adolfo Gustavo Serra Seca Neto, Dr.**
presidente - (UTFPR - CT)

Prof. **Andrey Ricardo Pimentel, Dr.**
(UFPR)

Prof. **Maria Cláudia Figueiredo P. Emer, Dra.**
(UTFPR - CT)

Prof. **Robinson Vlda Noronha, Dr.**
(UTFPR - CT)

Dedico este trabalho à minha família, amigos, categoria onde incluo meus orientadores e também às pessoas que desenvolvem e colaboram com software livre. Todo esse trabalho foi produzido com software livre e sem a dedicação dessas pessoas que eu nem conheço pessoalmente pude aprender muito e realizar um bom trabalho. Dedico especialmente a John Hunter in memoriam , autor da biblioteca matplotlib que faleceu em 2012.

AGRADECIMENTOS

A realização de um trabalho como uma dissertação de mestrado exige muito tempo e dedicação, mas não é uma luta que se luta só. Em alguns momentos é necessário enfrentar alguns obstáculos por conta própria. Mas a vitória só é conquistada com o apoio de várias pessoas e é portanto uma vitória coletiva. Muitos me deram os parabéns, estendo aqui a congratulação a todos que me ajudaram, pois conseguiram me tornar uma pessoa melhor e ainda mais dedicada.

Em primeiro lugar agradeço aos meus pais, Jaime e Maria Ângela. Sou o que sou, cheguei aonde cheguei, graças à dedicação de minha mãe e ao esforço de meu pai. Sempre me lembro da história que ele contava sobre a infância, indo pescar com alfinetes tortos, pois não tinham dinheiro para comprar anzol.

Agradeço minha esposa e meu filho, Nerli e Miguel, que precisaram abrir mão de minha companhia devido à compromissos do mestrado, entre fins de semana trabalhando na dissertação e viagens para congressos. Palavras de apoio e incentivo nunca faltaram, otimismo e confiança me deram forças para vencer esse desafio.

Agradeço aos meus orientadores, Adolfo e Robinson, que souberam ser verdadeiros amigos. Não aqueles que falam o que você quer ouvir, mas aqueles que ouvem e aconselham nas horas de desespero e falam o que é necessário, mesmo não sendo o que você quer ouvir. Esse é o tipo de amizade valiosa e profunda pois mostra preocupação.

Agradeço por fim aos meus amigos próximos e distantes que colaboraram com a pesquisa, respondendo questionários, entrevistas e por vezes simplesmente ouvindo meus desabafos. Representando todos os amigos que esquecerei de listar, menciono: Alexandre Freire, Daniel Cukier, Daniel Wildt, Danilo Sato, Elizabeth Leddy, Hugo Corbucci, Mauricio Aniche, Roberto Rodrigues. Agradecimentos especiais para Rodrigo Senra e Marcel Caraciolo, com suas dicas pude melhorar a qualidade do conteúdo deste trabalho. Muitas outras pessoas me ajudaram de uma forma ou de outra, peço que me desculpem por não mencionar seus nomes aqui, mas me sinto agradecido por todo apoio.

Há homens que lutam um dia e são bons,
há outros que lutam um ano e são
melhores, há os que lutam muitos anos e
são muito bons, mas há os que lutam toda
a vida e estes são imprescindíveis.

Bertolt Brecht

RESUMO

LUZ, Ramiro. A influência do Dojo de Programação no ensino de práticas ágeis. 69 f. Dissertação – Programa de Pós-graduação em Computação Aplicada, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Dojo de Programação é uma atividade dinâmica e colaborativa inspirada em artes marciais onde é possível praticar programação, especialmente técnicas relacionadas a métodos ágeis. Os métodos de ensino atuais tratam todos os alunos da mesma forma. Cada pessoa tem uma história, experiências, habilidades, conhecimentos. Raramente as particularidades de cada indivíduo são respeitadas. Buscamos verificar qual a influência do Dojo de Programação como atividade de ensino a fim de promover a interação entre os alunos. O Dojo de Programação oferece os recursos necessários para ensino acadêmico de programação de computadores? Para responder a questão foram realizados questionários com leigos e participantes ativos de Dojo de Programação, participantes espontâneos de grupos de Dojo de Programação e participantes compulsórios de alunos que participaram de Dojo de Programação em aula de disciplina regular de especialização na Universidade Tecnológica Federal do Paraná. Foram realizadas entrevistas com especialistas organizadores de encontros de Dojo de Programação. Pontos positivos, como a participação dos alunos e negativos, como o fato de não ser uma atividade adequada para apresentação de assuntos teóricos, foram ponderados. Os resultados obtidos sugerem que a atividade pode ser utilizada de forma complementar em disciplinas de programação de computadores a fim de aumentar a participação dos alunos e permitir ao professor conhecer as dificuldades e facilidades de cada aluno individualmente.

Palavras-chave: Dojo de Programação, TDD, Programação Extrema XP, ensino de programação, métodos ágeis, programação em pares, ambiente de aprendizado interativo, aprendizado colaborativo, aprendizado baseado em problemas

ABSTRACT

LUZ, Ramiro. Coding Dojo influence on teaching agile practices. 69 f. Dissertação – Programa de Pós-graduação em Computação Aplicada, Universidade Tecnológica Federal do Paraná. Curitiba, 2013.

Coding Dojo is a dynamic and collaborative activity inspired by martial arts where you can practice programming, especially techniques related to agile methods. The existing teaching methods treat all students the same way. Each person has a story, experiences, skills, knowledge. Rarely the particularities of each individual are respected. The purpose of this research is to verify the influence of the Coding Dojo in teaching agile practices. The Coding Dojo offers the required resources to academic teaching computer programming? To answer the question surveys were conducted with novice and active participants of Coding Dojo, also spontaneous participants in groups of Coding Dojo and students who attended compulsorily in a Coding Dojo during a class of a regular discipline of one specialization course in Federal Technological University of Paraná. Interviews were conducted with experts organizers of meetings of Coding Dojo. Strengths, such as student participation and weaknesses, as not being a suitable practice for presenting theoretical contents were considered. The obtained results suggest that the activity can be used complementary in computer programming disciplines to increase student participation and allow the teacher to know the difficulties and facilities of each student individually.

Keywords: Coding Dojo, TDD, XP, interactive learning environment, teaching programming, agile methods, pair programming, collaborative learning, problem based learning

LISTA DE FIGURAS

FIGURA 1	– FOTO DE UM DOJO DE PROGRAMAÇÃO	16
FIGURA 2	– FREQUÊNCIA DE RESPOSTAS DO QUESTIONÁRIO ELETRÔNICO - AMBIENTE	34
FIGURA 3	– FREQUÊNCIA DE RESPOSTAS DO QUESTIONÁRIO ELETRÔNICO - INFRA-ESTRUTURA	36
FIGURA 4	– FREQUÊNCIA DE RESPOSTAS DO QUESTIONÁRIO ELETRÔNICO - APRENDIZADO	38
FIGURA 5	– EXPERIÊNCIA EM DOJO DE PROGRAMAÇÃO.	40
FIGURA 6	– CONHECIMENTO SOBRE DOJO DE PROGRAMAÇÃO.	41
FIGURA 7	– FREQUÊNCIA DE RESPOSTAS DO GRUPO 1	43
FIGURA 8	– FREQUÊNCIA DE RESPOSTAS DO GRUPO 2	44

LISTA DE TABELAS

TABELA 1	– RESPOSTAS DO QUESTIONÁRIO ELETRÔNICO - AMBIENTE ...	33
TABELA 2	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO SOBRE O AMBIENTE - INICIANTES	33
TABELA 3	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO SOBRE O AMBIENTE - EXPERIENTES	34
TABELA 4	– RESPOSTAS DO QUESTIONÁRIO ELETRÔNICO - ESTRUTURA .	35
TABELA 5	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO SOBRE INFRA-ESTRUTURA - INICIANTES	35
TABELA 6	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO SOBRE INFRA-ESTRUTURA - EXPERIENTES	35
TABELA 7	– RESPOSTAS DO QUESTIONÁRIO ELETRÔNICO - APRENDIZADO	37
TABELA 8	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO DO APRENDIZADO - INICIANTES	37
TABELA 9	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO DO APRENDIZADO - EXPERIENTES	37
TABELA 10	– CARACTERÍSTICAS DO DOJO DE PROGRAMAÇÃO	39
TABELA 11	– FREQUÊNCIA DE RESPOSTAS DO GRUPO 1	42
TABELA 12	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO DO QUESTIONÁRIO SOBRE O DOJO - GRUPO 1	42
TABELA 13	– FREQUÊNCIA DE RESPOSTAS DO GRUPO 2	43
TABELA 14	– MÉDIA, VARIÂNCIA E DESVIO PADRÃO DO QUESTIONÁRIO SOBRE O DOJO - GRUPO 2	44
TABELA 15	– COMPARAÇÃO DO PERCENTUAL DO QUESTIONÁRIO PRESENCIAL	48

LISTA DE SIGLAS

TDD	Test Driven Development
XP	Extreme Programming
IME	Instituto de Matemática e Estatística
EACH	Escola de Artes, Ciências e Humanidades
CCSL	Centro de Competência em Software Livre

SUMÁRIO

1	INTRODUÇÃO	13
1.1	MOTIVAÇÃO E JUSTIFICATIVA	19
1.2	OBJETIVOS	20
1.2.1	Objetivo Geral	20
1.2.2	Objetivos Específicos	20
2	REVISÃO BIBLIOGRÁFICA	21
2.1	PESQUISAS QUE ENVOLVEM MÉTODOS ÁGEIS, ENSINO E EDUCAÇÃO	21
2.2	PUBLICAÇÕES RELACIONADAS A DOJO DE PROGRAMAÇÃO	25
2.3	CONTRIBUIÇÃO COM OS TRABALHOS RELACIONADOS	27
3	MÉTODO DE PESQUISA	28
3.1	QUESTIONÁRIO ELETRÔNICO	28
3.2	ENTREVISTAS	29
3.3	QUESTIONÁRIOS PRESENCIAIS	29
4	RESULTADOS	30
4.1	DADOS DO QUESTIONÁRIO ELETRÔNICO	31
4.1.1	Questões sobre o ambiente e técnicas usadas	32
4.1.2	Questões sobre a infra-estrutura e recursos utilizados	34
4.1.3	Questões sobre a percepção de aprendizado	36
4.2	ENTREVISTAS	38
4.3	DADOS DOS QUESTIONÁRIOS PRESENCIAIS	39
4.4	COMENTÁRIOS SOBRE OS DADOS	44
5	DISCUSSÃO DOS RESULTADOS	46
5.1	QUESTIONÁRIO ELETRÔNICO	46
5.2	ENTREVISTAS	47
5.3	QUESTIONÁRIO PRESENCIAL	47
5.4	OBSERVAÇÕES	48
6	CONCLUSÃO	49
	REFERÊNCIAS	51
7	APÊNDICE: QUESTIONÁRIOS E ROTEIRO DA ENTREVISTA COM ESPECIALISTAS	54
7.1	QUESTIONÁRIO ABERTO À COMUNIDADE.	54
7.2	QUESTIONÁRIO PRESENCIAL SOBRE O PERFIL DOS PARTICIPANTES.	57
7.3	QUESTIONÁRIO PRESENCIAL SOBRE A OPINIÃO DOS PARTICIPANTES.	59
7.4	QUESTÕES DAS ENTREVISTAS	60
7.5	RESUMO DAS ENTREVISTAS COM ESPECIALISTAS EM DOJO DE PROGRAMAÇÃO	60
7.5.1	Alexandre Freire	61
7.5.2	Daniel Cukier	62
7.5.3	Daniel Wildt	64
7.5.4	Danilo Sato	65
7.5.5	Elizabeth Leddy	65

7.5.6 Hugo Corbucci	66
7.5.7 Mauricio Aniche	67
7.5.8 Roberto Rodrigues	68
7.5.9 Considerações sobre as entrevistas	69

1 INTRODUÇÃO

Dojo, traduzido do Japonês, significa “lugar do caminho”. Este termo foi usado originalmente para designar o espaço de meditação dos monges Budistas. Neste contexto, pode ser traduzido como “lugar onde se estuda a vida”. Hoje em dia é conhecido como o lugar onde artes marciais são praticadas.

Praticantes de artes marciais conhecem bem o termo *Kata*. *Kata* é uma sequência de exercícios repetidos diversas vezes com uma pequena melhoria a cada exercício. Em seu *blog*¹, Dave Thomas apresenta a ideia dos *Katas* de código (THOMAS, 2007a), percebendo a necessidade que os programadores tem de praticar assim como atletas e artistas fazem. Músicos, por exemplo, repetem exaustivamente a mesma música em busca de aperfeiçoamento; são exercícios planejados para serem executados com o objetivo de aprender e reforçar conceitos durante a prática, sem o compromisso de atingir uma única resposta correta.

Diferente dos problemas propostos em maratonas de programação onde o foco é a competição, como o Concurso Internacional de Programação Universitário da ACM², os *Katas* de código são usados como instrumento para o exercício colaborativo. Eventualmente podem ser usados desafios semelhantes aos encontrados em (SKIENA; REVILLA, 2003), entretanto é importante reforçar a ideia de que *Katas* de código ou desafios, usados durante um Dojo de Programação são problemas simples que servem como ferramenta para o aprendizado colaborativo e inclusivo, a competição é desaconselhada durante o Dojo de Programação.

Existem algumas variações na maneira de conduzir um Dojo de Programação: o estilo *Prepared Kata*, o estilo *Randori Kata* e o estilo *Kake Kata*.

¹<http://codekata.pragprog.com/>

²ACM International Collegiate Programming Contest - <http://icpc.baylor.edu/>

O estilo chamado de *Prepared Kata* consiste na apresentação simulada de um Dojo de Programação, normalmente realizada em eventos para divulgar o Dojo de Programação para plateias relativamente grandes. Permite a interrupção a qualquer momento para esclarecimento de dúvidas da audiência. Também permite que os interessados reproduzam os passos posteriormente.

O estilo mais difundido e utilizado, *Randori Kata*, é iniciado pela escolha de um problema, desafio ou *Kata*. O *Kata* é usado como meio para exercitar algumas práticas ágeis como programação em par, desenvolvimento dirigido por testes e passos de bebê em um único computador ligado à um projetor. Possui um mecanismo para troca da dupla, existem diversos mecanismos, mas o mais utilizado é chamado de *TimedBox*. A cada intervalo curto de tempo, entre 5 e 7 minutos por exemplo, o piloto volta para a plateia, o co-piloto assume o teclado e uma pessoa da plateia assume o papel de co-piloto. Deve sempre começar do zero, mesmo quando um problema já tiver sido usado em um encontro, tudo o que foi feito deve ser ignorado e uma nova solução deve ser iniciada. Devem ser usadas as práticas de desenvolvimento dirigido por testes e passos de bebê. É esperado de todos os participantes que prestem atenção ao que está sendo feito. O par deve explicar constantemente o que está fazendo permitindo que todos acompanhem. A audiência pode fazer sugestões quando os testes estiverem passando, esse estado é chamado de sinal verde. A dupla não deve ser interrompida quando os testes não estão passando, esse estado é chamado de sinal vermelho. A dupla pode pedir ajuda para a audiência. O Dojo acaba quando esgotar o tempo, normalmente após duas ou três horas de trabalho. Deve ser reservado cerca de 20 minutos para realização de uma retrospectiva da atividade, destacando o que pode melhorar e o que foi bom.

No estilo *Kake Kata* seguem-se os mesmos princípios do estilo *Randori Kata*, com algumas diferenças. São usados diversos computadores, em cada computador existe ao menos uma variação. As duplas trabalham em problemas diferentes, com linguagens diferentes, ambientes diferentes, inclusive pode ocorrer de ser tudo diferente. Quem não participa de nenhuma dupla não deve atrapalhar e fica livre aguardando a troca de pares. Após o turno de tempo os pilotos devem voltar para a plateia, ou podem se tornar co-pilotos em outro computador. Os co-pilotos se tornam pilotos e integrantes da plateia podem se tornar co-pilotos. Ao final, da mesma forma que o estilo *Randori*, deve ser realizada uma retrospectiva.

Dave Thomas percebeu as vantagens de praticar sem pressão por prazo, através de pequenos passos incrementais, certo dia quando teve um tempo livre, enquanto esperava seu filho na aula

de karatê. Ao desafiar um problema sem compromisso percebeu que isso é uma boa maneira de praticar e se aperfeiçoar. Em outra conversa com um amigo sobre a prática de músicos, vislumbrou a possibilidade de adaptar a ideia para programadores, que raramente praticam fora do ambiente de trabalho. Dessa maneira foi idealizada a prática de programação com *Katas* de código (*Code Kata*) (THOMAS, 2007b). O princípio por trás dessa ideia é a prática deliberada (ERICSSON et al., 1993).

Segundo (ERICSSON et al., 1993), o nível máximo de performance para um indivíduo em um dado domínio não é obtido automaticamente em função de longa experiência mas o nível de performance pode ser aprimorado em indivíduos altamente experientes como resultado de esforços deliberados para melhorar. Afirma também que na ausência de *feedback* adequado, aprendizado eficiente é impossível e melhora apenas o mínimo mesmo para sujeitos altamente motivados. Em (ERICSSON et al., 1993), o autor explica que, considerando três tipos de atividades genéricas, trabalhar, jogar e prática deliberada. Trabalhar inclui apresentação pública, competições, serviços prestados para pagamento entre outras atividades motivadas diretamente por recompensas externas. Jogar inclui atividades que não possuem um objetivo explícito e são inerentemente agradáveis. Prática deliberada inclui atividades que foram especialmente projetadas para aprimorar o nível atual de performance. As metas, custos e recompensas dessas três atividades diferem assim como a frequência em que os indivíduos as procuram. Sugere então que a prática deliberada como uma atividade de esforço pode ser sustentada por um tempo limitado a cada dia durante períodos estendidos sem levar à exaustão. Para maximizar os ganhos da prática a longo prazo, indivíduos precisam evitar a exaustão e devem limitar a prática em uma quantidade que eles podem se recuperar completamente em uma base diária ou semanal.

Essas ideias deram origem aos Dojos de Programação (*Coding Dojo*), que são resumidamente encontros de programadores com a intenção de exercitar programação utilizando métodos ágeis. O Dojo de Programação fornece um ambiente apropriado para realização de esforço deliberado. Possui mecanismos que permitem ao participante fazer uma auto-avaliação e para que outros participantes possam dar o *feedback* necessário com o objetivo de melhorar continuamente. O Dojo de Programação possui atividades especialmente projetadas para aprimorar as habilidades dos programadores e é realizado com uma frequência semanal, quinzenal ou mensal, evitando assim a exaustão.

A atividade, Dojo de Programação, é disciplinada e pode ser melhor explicada fazendo-se uma analogia com as artes marciais. Durante a prática de artes marciais um instrutor, nor-

malmente chamado de *sensei*, organiza os discípulos em um círculo largo e então o *sensei* convida dois discípulos para o centro. Estes lutam seguindo as regras da modalidade de acordo com as orientações do *sensei*. Após alguns minutos de luta os discípulos saem do centro do círculo, sentam-se e o *sensei* convida outros dois discípulos para o centro. Dessa forma os discípulos que estão lutando aprendem praticando, os discípulos que estão no círculo aprendem com a observação e quanto mais lutarem e observarem melhor será o aprendizado.

No Dojo de Programação os participantes colaboram uns com os outros e todos aprendem juntos. Na Figura 1 é apresentada uma foto de um encontro de Dojo de Programação. De forma simplificada pode-se dizer que durante um encontro de Dojo de Programação, programadores se reúnem para resolver desafios de programação. Os desafios devem ser resolvidos utilizando práticas de métodos ágeis. Em um único computador os participantes se revezam em pares para resolver um desafio, sempre com o cuidado de explicar à todos os participantes o que está sendo feito. A tela do computador é projetada para que todos vejam o que está acontecendo.



Figura 1: Foto de um encontro de Dojo de Programação

Fonte: Autoria própria

Para realização de Dojos de Programação é necessário um projetor, um computador e um quadro para escrever ou desenhar algo relacionado à atividade. Preferencialmente a sala não deve possuir computadores para que todos prestem atenção no par que está programando e na solução projetada. Opcionalmente é feito um lanche após a atividade, oportunizando um momento de socialização e descontração onde discussões podem ser continuadas.

Cada encontro é dividido em etapas (GAILLOT, 2012). Podem ocorrer variação de grupo para grupo, mas o mais comum é seguir o processo sugerido pelo primeiro Dojo de Programação conhecido (BOSSAVIT; GAILLOT, 2005b).

A primeira coisa a ser feita é definir a data do próximo encontro, isso permite que os participantes se organizem para participar da atividade. Em alguns grupos os encontros ocorrem sempre no mesmo dia e horário. É sugerido tomar essa decisão em cerca de dois minutos.

O próximo passo é fazer uma retrospectiva da última sessão. Discute-se o que foi frustrante, o que foi interessante e o que foi bem. Essa etapa dura cerca de vinte e cinco a trinta minutos. Alguns grupos fazem a retrospectiva logo após o fim de cada encontro. O objetivo dessa etapa é fazer uma auto-avaliação para melhorar próximos encontros.

Então os participantes decidem qual será o tópico da sessão, qual problema e qual a linguagem de programação que será usada para abordar o problema. Com uma duração de aproximadamente dez minutos, essa etapa prioriza a escolha de uma tecnologia conhecida por ao menos um participante. Não é possível realizar a atividade se ninguém pode tirar dúvidas e aconselhar sobre as melhores técnicas e funcionamento das ferramentas disponíveis para a linguagem de programação escolhida.

Durante cerca de quarenta minutos os participantes programam, utilizando técnicas de desenvolvimento ágil e seguindo o estilo de Dojo de Programação adotado, *Prepared Kata*, *Randori Kata* ou *Take Kata*.

Opcionalmente pode ser feito um intervalo de cinco a dez minutos para refletir sobre o andamento e abordagem da solução, alguns grupos fazem essa reflexão ao longo do desenvolvimento e outros não fazem. Após o intervalo, o desenvolvimento continua por mais quarenta minutos.

Algumas técnicas de métodos ágeis recebem maior atenção, especialmente desenvolvimento dirigido por testes, programação em pares e passos de bebê.

Desenvolvimento dirigido por testes (TDD), é uma prática ágil baseada na produção de testes automatizados que verificam as funcionalidades de um software. O processo de testar e criar

funcionalidades guia o projeto e desenvolvimento. Para cada pequena parte da funcionalidade, um teste que descreve e verifica aquilo que o código deve fazer é escrito. Então é produzido apenas o código necessário e suficiente para que o código passe no teste. Em seguida é realizada a refatoração, termo usado para melhorias que simplificam e tornam o código mais claro, do código da funcionalidade e do teste. Segundo (BECK, 2002), isso permite que sejam escritos programas confiáveis, com testes unitários e com projetos simples para problemas complexos. Os testes fornecem feedback constante sobre o funcionamento do código além de proporcionar segurança para alterações. Quando algo precisa ser alterado é possível realizar a alteração e executar os testes automatizados, que irão verificar tudo que já estava funcionando anteriormente. Com isso o desenvolvedor não precisa ficar avaliando o código para saber se uma mudança vai afetar outras partes do software que já estavam corretas. A sequência de desenvolvimento utilizada em TDD é a seguinte:

- Primeiro, escreve-se um teste simples para uma pequena etapa da solução do problema.
- Deve-se executar os testes para ver o que falhou, algumas ferramentas já fazem isso automaticamente.
- Em seguida escreve-se o código mínimo necessário para que esse teste esteja correto.
- Roda-se os testes para confirmar o funcionamento, o termo mais conhecido pelos participantes é: o código passou nos testes.
- Então se o código passou nos testes, pode-se melhorá-lo ou, em outros termos, refatorá-lo.
- E por fim, escreve-se um novo teste para uma nova etapa voltando ao início.

Durante o ciclo de desenvolvimento dirigido por testes outra prática ágil, chamada de passos de bebê, recomenda que se mantenha as coisas simples, sem avançar muito de cada vez (BECK; ANDRES, 2004). Essa prática proporciona um avanço gradual em pequenos passos, aumentando assim as chances de que o próximo passo seja trivial (BOSSAVIT; GAILLOT,).

A Programação em par, uma das práticas ágeis mais importantes no Dojo de Programação, consiste no compartilhamento de um mesmo computador por dois desenvolvedores. O par colabora continuamente no mesmo projeto, algoritmo, código ou teste. Essa prática incentiva a comunicação constante e uma das suas principais vantagens é a revisão do código instantânea. Um dos membros do par, chamado de piloto, utiliza o teclado para codificar. O outro integrante do par, chamado de co-piloto, observa o trabalho do piloto procurando por erros de sintaxe,

digitação, chamada da função ou procedimento errado e assim por diante, (WILLIAMS; KESSLER, 2002).

Nos Dojos de Programação os participantes exercem papéis. Um dos participantes é o Sensei, que organiza a atividade. A plateia observa a evolução do código e o par desenvolve o código praticando a programação pareada, um deles como piloto e o outro como co-piloto. Devem ser aplicadas práticas de métodos ágeis, TDD, refatoração, passos de bebê (BECK; ANDRES, 2004) e (BOSSAVIT; GAILLOT,) e a programação em par. Desta forma e ao longo do tempo os participantes vão adquirindo familiaridade com as técnicas de TDD e aprendendo uns com os outros de forma colaborativa e evolutiva num ambiente amigável (SATO et al., 2008).

A presente pesquisa consistiu na realização de questionários com participantes de Dojo de Programação com diferentes realidades, alguns mais experientes, alguns voluntários e alguns alunos que participaram compulsoriamente durante aulas regulares. Foram realizadas entrevistas com especialistas experientes na realização de Dojos de Programação.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

Conforme relatado em (PIMENTEL et al., 2003), há um baixo índice de assimilação dos estudantes em disciplinas que requerem o conhecimento de programação. Um dos motivos, segundo (PIMENTEL et al., 2003), é a dificuldade em lidar com as diferenças de origem, experiência e habilidades entre aprendizes. Os docentes e os métodos atuais nem sempre permitem identificar o problema de cada indivíduo. Estas dificuldades podem ser diagnosticadas pelo alto grau de repetência nas disciplinas introdutórias e pelas dificuldades demonstradas pelos estudantes nas disciplinas avançadas que exigem o pré-requisito de programação, afirma (PIMENTEL et al., 2003).

Buscamos então identificar qual a influência do Dojo de Programação como atividade de ensino a fim de promover a interação entre os alunos e auxiliar o professor a identificar as dificuldades dos alunos individualmente.

1.2 OBJETIVOS

O Dojo de Programação apresenta algumas características semelhantes ao ensino baseado em problemas (PRINCE, 2004) em um ambiente participativo e colaborativo. A possibilidade de utilizar essa atividade para ensino de práticas ágeis pautaram os objetivos que seguem.

1.2.1 Objetivo Geral

Verificar a influência do Dojo de Programação em ambientes de ensino de programação usando práticas ágeis.

1.2.2 Objetivos Específicos

Em acordo com o objetivo geral foram determinados os seguintes objetivos específicos:

- Obter a opinião de especialistas e leigos sobre a prática do Dojo de Programação.
- Identificar as vantagens e desvantagens do Dojo de Programação como atividade de ensino.
- Comparar as respostas obtidas de grupos diferentes e verificar a percepção, positiva ou negativa dos participantes sobre a atividade.

2 REVISÃO BIBLIOGRÁFICA

Em busca de trabalhos semelhantes o autor realizou um levantamento bibliográfico utilizando as ferramentas de procura nos principais bancos de dados de artigos acadêmicos filtrando artigos da área de ciências exatas. Foram selecionadas publicações que avaliaram o ensino de métodos ágeis ou usando métodos ágeis no ensino bem como publicações relacionadas a Dojo de Programação.

2.1 PESQUISAS QUE ENVOLVEM MÉTODOS ÁGEIS, ENSINO E EDUCAÇÃO

No artigo (SCHNEIDER, 2005), os autores examinaram os objetivos educacionais para educação de graduandos em Engenharia de Software e como estes objetivos poderiam ser atendidos com práticas de Programação Extrema (XP). Os autores apresentam a questão: É possível tornar o ensino de Engenharia de Software mais orientado à prática e abordar o equívoco de que a preocupação principal da Engenharia de Software é produzir um documento após o outro? É apresentada uma discussão sobre a Programação Extrema num ambiente educacional. Segundo os autores a Programação Extrema (XP), apesar de possuir muitos aspectos positivos alguns de seus princípios são insuficientes para as necessidades de aprendizado da maioria dos estudantes. Concluem afirmando que a Programação Extrema (XP) possui valor limitado para educação sobre desenvolvimento de sistemas de larga escala. Recursos limitados em muitas instituições tornam questionável se a infra-estrutura necessária pode ser configurada a fim de aplicar a Programação Extrema (XP) adequadamente. Enquanto algumas práticas são adequadas para o ambiente educacional outras contradizem os objetivos educacionais. Particularmente o foco no código torna difícil introduzir conceitos que são necessários para o desenvolvimento de sistemas de larga escala. Por outro lado algumas práticas da Programação Extrema (XP) podem ser úteis para ensino de desenvolvimento de sistemas de baixa escala, a Programação em Pares é um exemplo.

Durante um curso de engenharia de software na faculdade Meredith foi realizado um experimento por Barrett Koster (KOSTER, 2006). Durante o experimento os alunos perceberam os benefícios da programação em par e puderam aprender em um ambiente colaborativo e coletivo ao invés do tradicional ambiente competitivo e individualista. Através da possibilidade ou necessidade de compartilhar código aprenderam a criar código bem documentado, encapsulado adequadamente, de forma a possibilitar a integração com o código dos outros alunos. Conceitos de boas práticas de programação foram reforçados: orientação a objetos, organização de arquivos e versionamento, comentários bem escritos, padronização, nomenclatura e modularização são alguns exemplos que foram percebidos na prática. Outro ponto positivo, relacionado a TDD, foi que os estudantes perceberam que os testes ajudaram a definir claramente quais eram as metas que deveriam ser atingidas assim que os casos de testes foram escritos.

As interações permitiram aos estudantes evoluir o código aos poucos tendo partes funcionando ao longo das fases, o que diminuiu a frustração e sentimento de fracasso muitas vezes visto em cursos de ciência da computação. Os autores concluíram que os métodos ágeis ajudaram a melhoria do ensino de Engenharia de Software.

Em classes anteriores os estudantes reclamavam de seus companheiros de time e muitos alunos tinham programas funcionando precariamente. A preocupação era apenas com a aprovação. O contraste foi óbvio: Todos possuíam programas em funcionamento e havia um sentimento de camaradagem no laboratório. Segundo os autores, foi a primeira vez que os projetos possuíam testes completos. Relatam ao final que os métodos ágeis ajudaram a melhorar o ensino de Engenharia de Software.

Os métodos ágeis de uma forma geral são centrados em pessoas e em seu comportamento. No estudo realizado por (MELNIK; MAURER, 2005) foram avaliados estudantes de graduação convidados voluntariamente. Estes responderam questionários sobre suas opiniões e experiências com métodos ágeis. Foram encontrados indícios de que os estudantes tem um grande entusiasmo quanto às práticas ágeis. Indícios de desenvolvimento de qualidades profissionais foram percebidos em integrantes de times que trabalham com métodos ágeis, como comunicação, comprometimento, cooperação e adaptação. Os autores afirmam que trabalhos futuros podem ser realizados para validar os resultados.

No artigo (MILLER; SMITH, 2007) foi avaliado o uso de desenvolvimento dirigido a testes em classes de programação embarcada. O ensino de programação embarcada é, por diversas

razões, complexo. Os ambientes de desenvolvimento são diferentes daqueles a que os estudantes estão acostumados. James Miller e Michael Smith sugerem em seu artigo (MILLER; SMITH, 2007) que a utilização de TDD para ensinar estudantes iniciantes em desenvolvimento embarcado pode ser uma excelente ferramenta para combater essa complexidade. Os autores acreditam que os instrutores precisam adaptar os mecanismos de ensino para diminuir as dificuldades, propondo a adoção de TDD como componente central desse processo, dando as seguintes razões:

- Especificação pelo exemplo;
- *Feedback* automático;
- Especificação e desenvolvimento incremental;
- Abordagem formal para testes;
- Especificações não funcionais;
- Testes não funcionais;
- Bases sólidas para refatorar requisitos não funcionais.

Os autores receberam uma avaliação positiva de alguns estudantes. Os estudantes apreciaram a habilidade de usar os testes existentes para verificar que o equipamento está funcionando fora das horas regulares de laboratório, enquanto outros demonstraram falta de interesse em continuar com TDD. Os estudantes que avaliaram positivamente queriam saber como incentivar outros professores a adotarem TDD. O artigo aborda a experiência dos autores; no momento da publicação do artigo, o trabalho ainda estava em andamento.

Em outro trabalho, um estudo de caso real, estudantes aprenderam práticas ágeis através de um projeto gerenciado usando o método Scrum (BRUEGGE et al., 2009). Os autores relatam que o projeto foi um sucesso e exemplo de aplicação de práticas e princípios ágeis, mostrando como ideias ágeis podem ser ensinadas em ambientes acadêmicos. Mencionaram ainda que os estudantes se divertiram durante o aprendizado e desenvolvimento do projeto.

No artigo (VODDE; KOSKELA, 2007), os autores relatam sua experiência ao criar cursos de desenvolvimento dirigido por testes na Nokia Networks. Segundo os autores, o primeiro curso foi essencialmente uma série de palestras que não apresentou o impacto esperado. Então

revisaram o curso e adaptaram um exercício para contar linhas de código descrito nos *Katas de Código* de Dave Tomas (THOMAS, 2001). Os autores afirmam que, ao seguir o desenvolvimento dirigido por testes e refatorando o código, precisaram encarar os fatos e tomar uma atitude quanto a arquitetura pobre. Concluíram que o exercício de contar linhas é extremamente útil em fornecer percepções sobre desenvolvimento dirigido por testes e seus benefícios, incluindo:

- Remoção de dependências externas ajuda a melhorar a testabilidade.
- Pensamento refletivo promove a melhoria da arquitetura da solução.
- Uma arquitetura bem elaborada e boa cobertura de testes também ajudam no surgimento de novas arquiteturas.

Concluem comentando que a lição aprendida ao conduzir o curso com este exercício simples é apenas uma das que eles aprenderam.

Em (SINIAALTO; ABRAHAMSSON, 2007) foram avaliados projetos desenvolvidos usando TDD e foram obtidos indícios de que TDD não produz sistemas com alta coesão quando utilizados por desenvolvedores amadores. Também foi observado que TDD não produz código coeso automaticamente. O simples fato de usar TDD não é sinônimo de qualidade: é necessário treinamento.

Uma das atividades mais importantes do Dojo de Programação é a Programação em Par. Foram encontradas pesquisas sobre Programação em Par usadas em cursos de graduação como segue.

No artigo (BALIJEPALLY et al., 2009), os autores relatam a pesquisa realizada com estudantes para avaliar a eficiência de programação em pares. Alguns estudantes trabalharam em pares e outros individualmente, mas com a performance medida em duplas nominais, pares de alunos programando separadamente. Os estudantes resolveram tarefas introdutórias de níveis alto e baixo de complexidade. Foram medidos a qualidade do software, a satisfação e confiança na performance.

Os autores apresentam a questão “Programação em pares é melhor que programação individual?”. O objetivo de estudar a eficácia de programação em pares comparado a performance

efetiva de programadores individuais. Para tanto, realizaram um experimento controlado em laboratório: estudantes trabalharam em tarefas de nível introdutório, onde foram manipulados dois fatores: modo de programação (pares versus indivíduo) e dificuldade da tarefa (alta e baixa complexidade). Participantes individuais foram combinados em duplas nominais. Foram comparadas as performances dos pares colaborativos com as duas melhores duplas nominais.

Uma combinação de resultados formam a base para a resposta à questão: qualidade do software, satisfação e confiança na performance. As limitações relatadas foram: a experiência dos participantes relatada foi de 2 anos ou menos (38% relataram 2 anos de experiência); os estudantes poderiam não lembrar da sintaxe da linguagem de programação (foi fornecido acesso a documentação online); a diferença entre local de trabalho e o ambiente do laboratório, entre outros fatores. Segundo os autores a performance da programação em pares não excede a performance do seu melhor membro trabalhando individualmente. Nenhum efeito evidente foi observado quanto à complexidade das tarefas. Pares sentiram-se mais satisfeitos e confiantes na sua performance.

2.2 PUBLICAÇÕES RELACIONADAS A DOJO DE PROGRAMAÇÃO

Em (SATO et al., 2008), são apresentados os conceitos do Dojo de Programação, as lições aprendidas durante as reuniões semanais na Universidade de São Paulo - USP e discutem os aspectos do Dojo de Programação que promovem o aprendizado e compartilhamento de conhecimento.

No artigo (ANICHE; SILVEIRA, 2011) foi avaliado o aprendizado em ambientes cujas equipes utilizam métodos ágeis. Através de entrevistas semi-estruturadas com os colaboradores foram avaliadas as opiniões sobre o aprendizado na empresa onde os autores trabalham. Uma das atividades foi o Dojo de Programação, segundo os colaboradores não percebiam evolução devido aos passos de bebê (BECK; ANDRES, 2004), possivelmente causado pelo fato do time aprender com facilidade e rapidamente. Alternativas foram experimentadas, mudando o formato do Dojo de Programação, ao invés de usar o formato *Randori Kata*, usaram o formato *Prepared Kata* e *Kake Kata*. O que foi um consenso entre os colaboradores foi o fato de aprender trabalhando com programação em pares, mencionado como um meio de aumentar o aprendizado.

As técnicas de métodos ágeis, assim como outras técnicas de programação, são ensinadas na prática. Acredita-se que pela natureza da atividade essa é a melhor maneira de ensinar e apren-

der. No artigo (BRAVO; GOLDMAN, 2010), é apresentada a avaliação de dois métodos para ensino de métodos ágeis. Foram avaliados o Dojo de Programação, através de questionários os participantes sugerem que a atividade é muito efetiva, o outro método usou ferramentas automatizadas, que mostraram pouca influencia no ensino de métodos ágeis. O artigo foi baseado em sua dissertação (BRAVO, 2011). Os participantes responderam questionários on-line sobre a percepção de aprendizado voluntariamente. Foram encontrados indícios de que a prática é uma boa técnica de ensino aos que pouco conhecem ou tem pouca experiência com TDD, não ficando claro o efeito sobre participantes mais experientes.

Laurent Bossavit e Emmanuel Gaillot apresentaram um Workshop sobre Dojo de Programação (BOSSAVIT; GAILLOT, 2005a). Durante o Workshop foram recriados os mecanismos do Dojo de Programação para que os participantes experimentassem a atividade e eventualmente fossem inspirados a iniciar seus próprios Dojos de Programação. Emily Bache apresentou Workshop semelhante com uma pequena variação em (BACHE, 2009).

Em (CARMO; BRAGANHOLO, 2012) os autores estudaram diferentes formas de Dojo de Programação avaliando a aceitação dos alunos em cursos de graduação. Foram coletados relatos de alunos alegando problemas de motivação, timidez e insegurança como razões para não participar do Dojo de Programação. A dispersão da plateia causou incomodo durante a atividade em um dos formatos. Em outro formato a falta de comunicação foi relatada como problema. Os alunos mostraram preferencia pelas aulas de laboratório. Sugerem que a diferença entre um Dojo de Programação tradicional, onde os participantes comparecem voluntariamente e o Dojo adaptado, onde os alunos são obrigados a participar, é um fator que prejudica a motivação dos alunos. Ainda, houve uma distribuição de pontos para a equipe com o melhor desempenho, favorecendo a motivação dos alunos. Segundo os autores os alunos consideraram a primeira versão do Dojo de Programação, chamada de Dojos individuais, a pior abordagem, houve uma grande diversidade de opiniões. Pretendem avaliar os motivos de resistência por parte dos alunos em trabalho futuro.

Em outro estudo sobre o uso de Dojo de Programação em ambientes acadêmicos (DELGADO et al., 2012), os autores propõem adaptações do Dojo de Programação para utilização no ensino superior da área de computação de forma a criar novas formas de interação entre alunos e professores. Foram aplicados Dojos adaptados em 3 disciplinas distintas, os autores relatam os obstáculos encontrados e sugestões para contornar os obstáculos. Reuniram um conjunto de adaptações ao Dojo de Programação para aplicar em disciplinas de graduação. Al-

gumas características do Dojo de Programação são pertinentes para ensino de programação de computadores, mas na graduação muitas disciplinas usam a programação como instrumento para ensinar outros conceitos. Portanto o processo precisa ser adaptado e controlado para ser usado como atividade didática. Os autores consideraram as adaptações apresentadas efetivas, entretanto ainda estão em estágio de maturação. A superação da timidez dos alunos foi considerado um desafio. Compilar problemas para serem usados em sessões de Dojo de Programação foi sugerido como trabalho futuro.

2.3 CONTRIBUIÇÃO COM OS TRABALHOS RELACIONADOS

Esta pesquisa avaliou a opinião de participantes de Dojo de Programação, com diversos níveis de experiência, motivações diferentes e grupos diferentes. Foram avaliadas opiniões de especialistas através de entrevistas que permitiram obter dados qualitativos sobre a atividade. Os resultados foram apresentados na Seção 4.1, Seção 4.2 e Seção 4.3. É apresentada uma discussão dos resultados no Capítulo 5. Contribuindo com o conhecimento sobre a utilização da atividade como ferramenta para auxiliar o ensino de práticas ágeis.

3 MÉTODO DE PESQUISA

Durante o período da pesquisa, foram realizados treze encontros de Dojo de Programação (DOJOPR, 2012), sempre com o cuidado de registrar a ata dos encontros com a opinião dos participantes que destacaram pontos positivos e negativos de acordo com sua percepção. Essa retrospectiva serve para avaliar e aprimorar os encontros, foi usada na pesquisa para delinear os questionários e entrevistas. Um questionário eletrônico, Apêndice 7.1, disponibilizado na internet e divulgado nas listas de discussão de grupos de Dojo de Programação brasileiros foi respondido por participantes de Dojo de Programação com diferentes níveis de experiência. A intenção desse questionário foi obter a opinião de participantes regulares de Dojo de Programação sobre as práticas ágeis usadas nos encontros. Foram realizadas entrevistas semi-estruturadas, Apêndice 7.5, com oito profissionais experientes em Dojo de Programação que relataram suas opiniões sobre a prática da atividade, fornecendo para a pesquisa dados qualitativos sobre a atividade Dojo de Programação. Em seguida foram aplicados questionários, Apêndices 7.2 e 7.3, com alunos de especialização da Universidade Tecnológica Federal do Paraná e voluntários da comunidade Dojo Paraná que participaram de encontros de Dojo de Programação. O objetivo desses questionários foi obter informações de grupos distintos, um grupo de alunos que participaram compulsoriamente da atividade e um grupo de participantes voluntários, a fim de comparar as respostas.

3.1 QUESTIONÁRIO ELETRÔNICO

Foi disponibilizado um questionário eletrônico, Apêndice 7.1, respondido por voluntários participantes de Dojo de Programação de diversas regiões do país. Foram solicitadas informações sobre o perfil dos participantes para comparar as respostas dos participantes mais experientes com as respostas dos menos experientes, considerando experientes aqueles que participaram de mais de 20 encontros de Dojo de Programação. O questionário solicitou a opinião dos participantes com relação à prática de Dojo de Programação e seus efeitos no aprendizado de técnicas de desenvolvimento ágil. Uma escala de Likert foi usada para as questões de opinião.

3.2 ENTREVISTAS

Para melhor entender os aspectos subjetivos da técnica do Dojo de Programação, entrevistas semi-estruturadas, Apêndice 7.4, foram usadas para guiar o entrevistador e ao mesmo tempo permitir ao entrevistado expressar sua opinião. Segundo (KVALE, 1996), com entrevistas em pesquisas qualitativas você tenta entender algo do ponto de vista do sujeito e descobrir o significado de suas experiências.

3.3 QUESTIONÁRIOS PRESENCIAIS

Um questionário serviu para determinar o perfil dos participantes, Apêndice 7.2, com questões relacionadas a experiência profissional e conhecimento prévio de práticas de métodos ágeis como programação em pares, passos de bebê e desenvolvimento dirigido por testes. Outro questionário obteve a opinião dos participantes sobre a atividade do Dojo de Programação, Apêndice 7.3. As questões apresentadas estavam relacionadas à participação no encontro realizado logo antes do preenchimento do questionário. Os questionários foram submetidos a dois grupos distintos. Um dos grupos composto por alunos de curso de especialização que participaram compulsoriamente de um Dojo de Programação. O segundo grupo foi de voluntários que participaram de um encontro regular do grupo Dojo Paraná. O questionário sobre o perfil dos participantes foi apresentado antes da atividade e o questionário sobre a opinião dos participantes foi apresentado depois da atividade.

Os questionários presenciais foram realizados para auxiliar o entendimento da diferença de opiniões sobre o Dojo de Programação em grupos distintos, com motivação e experiência diferentes.

4 RESULTADOS

Os resultados obtidos são apresentados nesse capítulo. Foram obtidas 64 respostas ao questionário eletrônico, Apêndice 7.1, 8 respostas de entrevistas, Apêndice 7.5, 21 respostas de alunos de curso de especialização da Universidade Tecnológica Federal do Paraná e 11 respostas de participantes do encontro do grupo Dojo Paraná aos mesmos questionários, Apêndice 7.2 e Apêndice 7.3.

Para obter a opinião de participantes regulares de Dojo de Programação foi elaborado um questionário eletrônico que foi disponibilizado e enviado para diversas comunidades que organizam encontros de Dojo de Programação. O questionário identificou o perfil dos 64 voluntários que participaram. Aproximadamente 30% se declararam profissionais, pouco mais de 34% estudantes, quase 36% de profissionais que também estudam.

O questionário eletrônico obteve informações sobre três categorias de questões. Informações sobre o ambiente e técnicas utilizadas, sobre os recursos e infra-estrutura necessários para realizar a atividade e sobre práticas ágeis usadas nos encontros. Com essas informações foi possível avaliar a opinião de membros da comunidade de participantes regulares de encontros de Dojo de Programação sobre ambiente, recursos e práticas ágeis. Os resultados detalhados são apresentados na Seção 4.1.

Entrevistas com especialistas, com experiência em organização de encontros de Dojo de Programação, foram realizadas e coletaram informações descritivas. Permitiram avaliar aspectos subjetivos da atividade, apresentados na Seção 4.2.

Questionários presenciais idênticos, foram aplicados em dois grupos diferentes. As informações coletadas permitiram avaliar e comparar opiniões sobre a dinâmica e práticas ágeis de um grupo de participantes regulares de Dojo de Programação e um grupo de alunos de especialização da Universidade Tecnológica Federal do Paraná. O grupo de alunos participou

de forma compulsória durante uma das aulas e os participantes regulares o fizeram de forma voluntária. Resultados apresentados na Seção 4.1.

4.1 DADOS DO QUESTIONÁRIO ELETRÔNICO

Foram apresentadas questões que usaram a escala de Likert, com cinco opções, Discordo totalmente (-2), Discordo Parcialmente (-1), Neutro (0), Concordo parcialmente (1) e Concordo totalmente (2). Para melhorar a apresentação dos resultados foi criada a seguinte legenda:

- Q1 - Qual a categoria em que você se encaixa?
- Q2 - Qual sua idade?
- Q3 - Sabe programar à quanto tempo?
- Q4 - Gênero:
- Q5 - Você sabe o que é "Test Driven Development"?
- Q6 - Experiência com "Test Driven Development":
- Q7 - Experiência com programação em par:
- Q8 - Você sabe o que é programação em par?
- Q9 - Você sabe o que são passos de bebê ("Baby steps")?
- Q10 - Experiência com passos de bebê ("Baby steps"):
- Q11 - Há quanto tempo participa de dojos de programação?
- Q12 - Já participou de quantos encontros de dojo de programação aproximadamente?
- Q13 - Com que frequência participa de dojos de programação?
- Q14 - É melhor quando o intervalo de tempo entre os encontros é menor (por exemplo, semanal é melhor que quinzenal).
- Q15 - A prática utilizando linguagens de programação conhecida por todos resulta em melhor aprendizado.
- Q16 - Funciona bem quando todos os participantes são experientes.
- Q17 - Utilizar a técnica de programação em par atrapalha o andamento da atividade.

- Q18 - Passos de bebê dificultam o processo de encontrar uma solução para o problema.
- Q19 - Criar os testes antes do código ajuda o andamento da atividade.
- Q20 - Laboratórios com computadores são adequados para prática de dojo de programação.
- Q21 - É importante que o dojo de programação ocorra em um local fixo regularmente.
- Q22 - É fundamental que exista um quadro ou painel flip-chart para escrever.
- Q23 - Não é necessário ter um projetor se a dupla comunicar bem a ideia.
- Q24 - O dojo de programação favorece o aprendizado de novas linguagens de programação.
- Q25 - O dojo de programação dificulta o aprendizado de TDD.
- Q26 - O dojo de programação favorece o nivelamento dos participantes por conta da programação em par.
- Q27 - Praticar passos de bebê ensina a resolver problemas de forma gradual.
- Q28 - Durante a atividade de dojo de programação aprende-se bastante devido à troca de experiências.
- Q29 - Para aprender métodos ágeis prefiro treinamento tradicional, com exposição teórica e exercícios individuais.

As questões Q1 até Q13 solicitaram informações sobre o perfil dos participantes. As questões Q14 até Q29 solicitaram a opinião dos participantes usando a escala de Likert. As questões Q17, Q18, Q20, Q23, Q25, Q29 tiveram o sentido invertido para evitar influenciar os participantes a responder sempre de forma positiva.

4.1.1 Questões sobre o ambiente e técnicas usadas

As questões Q14, Q15, Q16, Q17, Q18, Q19 exibiram o seguinte texto explicativo para contextualizar a pergunta: “Com relação à atividade de dojo de programação estilo *Randori* (estilo mais usado, não precisa de preparação antecipada, participantes escolhem problema e/ou linguagem, possui mecanismo para troca de par, começa do zero):”, junto com o enunciado da questão. As questões dessa categoria estavam relacionadas ao ambiente, qual a periodicidade

preferida, nível de experiência dos participantes, técnicas e tecnologias utilizadas. A Tabela 1 contém a distribuição das frequências das respostas em quantidade e percentual desse grupo de questões.

Tabela 1: Frequência de respostas do questionário eletrônico sobre o ambiente do Dojo de Programação

	Discordo fortemente		Discordo parcialmente		Neutro		Concordo parcialmente		Concordo totalmente	
	cnt	perc	cnt	perc	cnt	perc	cnt	perc	cnt	perc
Q14	4	6.2%	6	9.4%	15	23.4%	20	31.2%	19	29.7%
Q15	5	7.8%	9	14.1%	13	20.3%	26	40.6%	11	17.2%
Q16	12	18.8%	22	34.4%	8	12.5%	14	21.9%	8	12.5%
Q17	40	62.5%	14	21.9%	7	10.9%	3	4.7%	0	0.0%
Q18	34	53.1%	13	20.3%	9	14.1%	8	12.5%	0	0.0%
Q19	2	3.1%	4	6.2%	9	14.1%	17	26.6%	32	50.0%

Na Tabela 2 são apresentados a média, variância e desvio padrão das respostas do questionário eletrônico dos participantes considerados iniciantes. Para ser considerado iniciante o participante deve ter respondido que participa de encontros há menos de 11 meses ou ter alegado participar de menos de 10 encontros de Dojo de Programação.

Tabela 2: Média, variância e desvio padrão do questionário eletrônico sobre o ambiente do Dojo de Programação - iniciantes

Variável	Média	Variância	Desvio padrão
Q14	0.58	1.20	1.08
Q15	0.72	1.21	1.09
Q16	-0.12	1.87	1.35
Q17	-1.19	0.92	0.95
Q18	-0.91	1.32	1.14
Q19	1.09	1.04	1.01

Na Tabela 3 são apresentados a média, variância e desvio padrão das respostas do questionário eletrônico dos participantes considerados experientes. Para ser considerado experiente o participante deve ter respondido que participa de encontros há mais de 1 ano e ter alegado participar de 10 ou mais encontros de Dojo de Programação.

Na Figura 2 é possível visualizar a frequência de respostas ao questionário eletrônico representando a tendência das opiniões dos participantes sobre o ambiente do Dojo de Programação.

Tabela 3: Média, variância e desvio padrão do questionário eletrônico sobre o ambiente do Dojo de Programação - experientes

Variável	Média	Variância	Desvio padrão
Q14	0.90	1.79	1.31
Q15	-0.10	1.29	1.11
Q16	-0.52	1.56	1.22
Q17	-1.90	0.09	0.29
Q18	-1.62	0.55	0.72
Q19	1.24	1.49	1.19

Os gráficos apresentam a frequência de respostas de todos os participantes, dos participantes experientes e inexperientes.

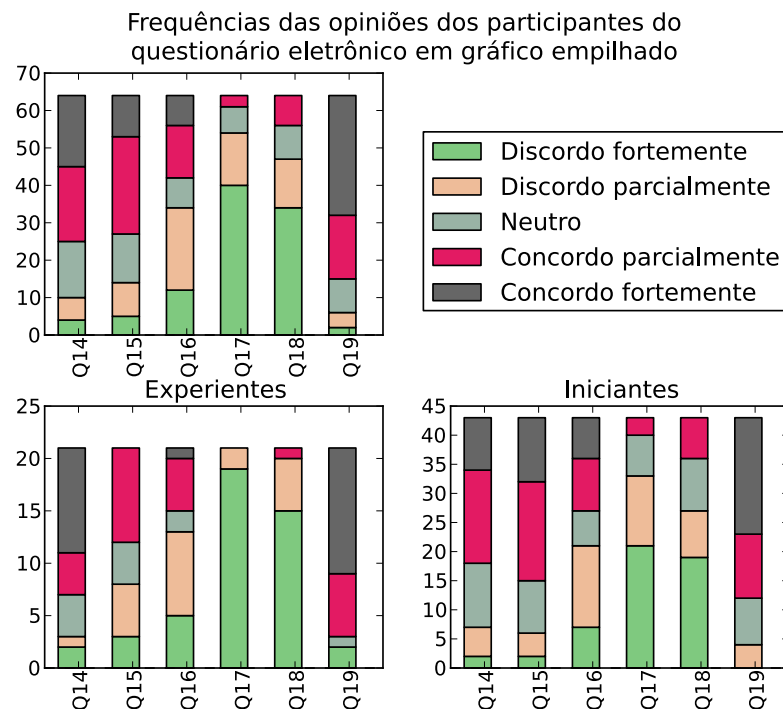


Figura 2: Frequência de respostas do questionário eletrônico sobre o ambiente do Dojo de Programação

4.1.2 Questões sobre a infra-estrutura e recursos utilizados

Já as questões Q20, Q21, Q22, Q23 exibiram o texto: “Com relação à estrutura física onde ocorre a prática de dojo de programação:”, junto com o enunciado da questão. Esta categoria estava relacionada à infra-estrutura e recursos necessários para a atividade. A Tabela 4 contém a distribuição das frequências das respostas em quantidade e percentual desse grupo de questões.

Tabela 4: Frequência de respostas do questionário eletrônico sobre a infra-estrutura do Dojo de Programação

	Discordo fortemente		Discordo parcialmente		Neutro		Concordo parcialmente		Concordo totalmente	
	cnt	perc	cnt	perc	cnt	perc	cnt	perc	cnt	perc
Q20	6	9.4%	13	20.3%	18	28.1%	11	17.2%	16	25.0%
Q21	4	6.2%	8	12.5%	17	26.6%	23	35.9%	12	18.8%
Q22	4	6.2%	3	4.7%	10	15.6%	22	34.4%	25	39.1%
Q23	30	46.9%	18	28.1%	4	6.2%	9	14.1%	3	4.7%

Na Tabela 5 são apresentados a média, variância e desvio padrão das respostas do questionário eletrônico dos participantes considerados iniciantes. Para ser considerado iniciante o participante deve ter respondido que participa de encontros há menos de 11 meses ou ter alegado participar de menos de 10 encontros de Dojo de Programação.

Tabela 5: Média, variância e desvio padrão do questionário eletrônico sobre a infra-estrutura do Dojo de Programação - iniciantes

Variável	Média	Variância	Desvio padrão
Q20	0.42	1.77	1.32
Q21	0.40	1.34	1.14
Q22	1.00	1.24	1.10
Q23	-0.95	1.52	1.22

Na Tabela 3 são apresentados a média, variância e desvio padrão das respostas do questionário eletrônico dos participantes considerados experientes. Para ser considerado experiente o participante deve ter respondido que participa de encontros há mais de 1 ano e ter alegado participar de 10 ou mais encontros de Dojo de Programação.

Tabela 6: Média, variância e desvio padrão do questionário eletrônico sobre a infra-estrutura do Dojo de Programação - experientes

Variável	Média	Variância	Desvio padrão
Q20	0.00	1.50	1.20
Q21	0.67	1.13	1.04
Q22	0.86	1.53	1.21
Q23	-1.05	1.65	1.25

Na Figura 3 é possível visualizar a frequência de respostas ao questionário eletrônico representando a tendência das opiniões dos participantes sobre a infra-estrutura e recursos utilizados

no Dojo de Programação. Os gráficos apresentam a frequência de respostas de todos os participantes, dos participantes experientes e inexperientes.

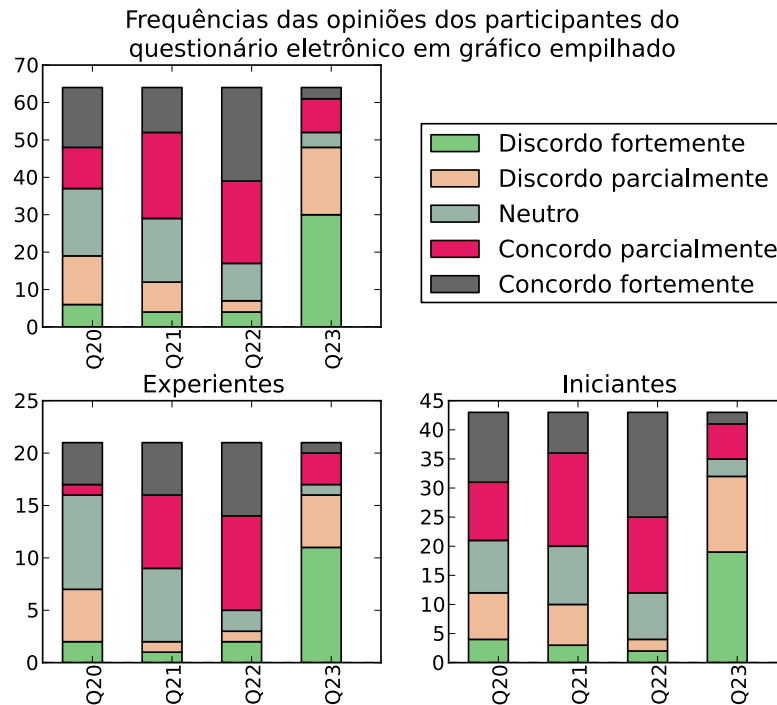


Figura 3: Frequência de respostas do questionário eletrônico sobre a infra-estrutura usada no Dojo de Programação

4.1.3 Questões sobre a percepção de aprendizado

As questões Q24, Q25, Q26, Q27, Q28, Q29 exibiram o texto: “Com relação à sua percepção do aprendizado durante a atividade de dojo de programação:”, junto com o enunciado da questão. Estas questões coletaram informações sobre a percepção dos participantes quanto ao aprendizado de diversas práticas ágeis. A Tabela 7 contém a distribuição das frequências das respostas em quantidade e percentual desse grupo de questões.

Na Tabela 8 são apresentados a média, variância e desvio padrão das respostas do questionário eletrônico dos participantes considerados iniciantes. Para ser considerado iniciante o participante deve ter respondido que participa de encontros há menos de 11 meses ou ter alegado participar de menos de 10 encontros de Dojo de Programação.

Na Tabela 9 são apresentados a média, variância e desvio padrão das respostas do questionário eletrônico dos participantes considerados experientes. Para ser considerado experiente

Tabela 7: Frequência de respostas do questionário eletrônico sobre o aprendizado de práticas de métodos ágeis usadas no Dojo de Programação

	Discordo fortemente		Discordo parcialmente		Neutro		Concordo parcialmente		Concordo totalmente	
	cnt	perc	cnt	perc	cnt	perc	cnt	perc	cnt	perc
Q24	1	1.6%	2	3.1%	2	3.1%	20	31.2%	39	60.9%
Q25	37	57.8%	11	17.2%	10	15.6%	4	6.2%	2	3.1%
Q26	0	0.0%	5	7.8%	10	15.6%	26	40.6%	23	35.9%
Q27	1	1.6%	3	4.7%	6	9.4%	11	17.2%	43	67.2%
Q28	1	1.6%	2	3.1%	0	0.0%	13	20.3%	48	75.0%
Q29	26	40.6%	14	21.9%	16	25.0%	4	6.2%	4	6.2%

Tabela 8: Média, variância e desvio padrão do questionário eletrônico sobre a percepção de aprendizado durante o Dojo de Programação - iniciantes

Variável	Média	Variância	Desvio padrão
Q24	1.40	0.82	0.89
Q25	-1.05	1.33	1.14
Q26	0.91	0.99	0.98
Q27	1.26	1.19	1.08
Q28	1.51	0.83	0.90
Q29	-0.70	1.69	1.29

o participante deve ter respondido que participa de encontros há mais de 1 ano e ter alegado participar de 10 ou mais encontros de Dojo de Programação.

Tabela 9: Média, variância e desvio padrão do questionário eletrônico sobre a percepção de aprendizado durante o Dojo de Programação - experientes

Variável	Média	Variância	Desvio padrão
Q24	1.62	0.45	0.65
Q25	-1.52	0.96	0.96
Q26	1.33	0.43	0.64
Q27	1.81	0.16	0.39
Q28	1.90	0.09	0.29
Q29	-1.14	0.93	0.94

Na Figura 4 é possível visualizar a frequência de respostas ao questionário eletrônico representando a tendência das opiniões dos participantes sobre a percepção de aprendizado de práticas ágeis no Dojo de Programação. Os gráficos apresentam a frequência de respostas de todos os participantes, dos participantes experientes e inexperientes.

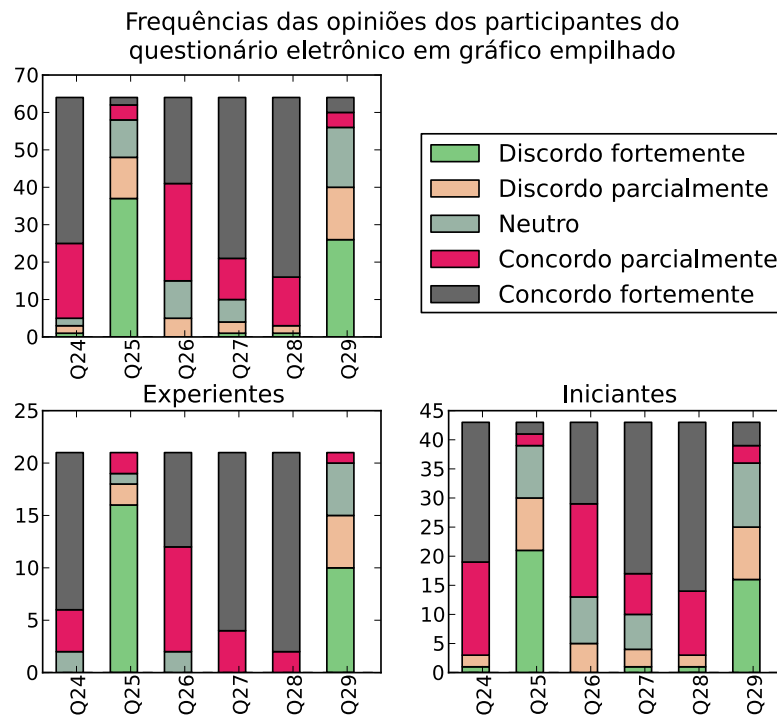


Figura 4: Frequência de respostas do questionário eletrônico sobre a percepção de aprendizado de práticas ágeis no Dojo de Programação

4.2 ENTREVISTAS

Especialistas com larga experiência em Dojo de Programação foram convidados a participar de entrevista sobre o Dojo de Programação. O critério adotado para o convite foi o fato de terem publicado artigo científico sobre Dojo de Programação ou por terem sido indicados diretamente por autor de artigo científico sobre Dojo de Programação. O resumo das entrevistas está disponível no Apêndice 7.5. As entrevistas revelaram a opinião dos especialistas com relação às vantagens e desvantagens da atividade, suas experiências pessoais e situações onde o Dojo pode ser empregado com mais e menos benefícios. Na Tabela 10 são apresentadas vantagens e desvantagens segundo a opinião dos entrevistados.

Os especialistas relataram que o Dojo de Programação aumentou a participação e interação entre os alunos em suas salas de aula. O Dojo de Programação também permitiu que as aulas seguissem um ritmo adequado à cada aluno. Segundo um dos especialistas, o Dojo de Programação facilita a elucidação de dúvidas graças ao aspecto colaborativo da atividade.

Por outro lado, vários entrevistados afirmaram que o Dojo de Programação deve intercalar suas atividades com apresentações de teoria visto que o Dojo de Programação tem um foco

Tabela 10: Características do Dojo de Programação segundo especialistas

Vantagens	Desvantagens
Ambiente Colaborativo e participativo. Aula dinâmica. Aumenta a comunicação.	Aulas expositivas apresentam a visão geral, mais abrangente de conceitos.
Indicado para atividades práticas. Foco na prática de conceito específico.	Foco em teoria, em conceitos.
Retrospectiva, processo crítico de melhoria contínua.	Pode sofrer preconceito quando realizado durante horário de trabalho.
Pode ser usado como ferramenta para avaliar candidatos a vagas de emprego.	Pode ter pouca participação quando realizado fora do horário de trabalho.
O ritmo é determinado pelo participante que está usando o teclado, chamado de piloto.	Grupos com participação frequente de novos membros evoluem muito lentamente.
Grupos assíduos evoluem gradualmente.	Ambiente expõe o participante.
Ajuda a perder a inibição. Socializa os participantes.	Pressão do tempo limite para troca de pares.
Ajuda a fixar a teoria. Proporciona oportunidade para tirar dúvidas pois todos estão observando.	Precisa ser complementado por aulas expositivas para apresentar teoria.
Fortalece a cultura de práticas ágeis.	
Estimula o trabalho em equipe.	
Mais interativo que aulas expositivas.	

muito grande na prática. Um dos especialistas afirmou que em alguns momentos o aprendiz precisa ficar sozinho, em silêncio e experimentar realizar as atividades por conta própria para fixar determinado conteúdo.

4.3 DADOS DOS QUESTIONÁRIOS PRESENCIAIS

Questionários presenciais foram aplicados em dois grupos distintos. Cada grupo respondeu dois questionários, um sobre o perfil dos participantes, outro sobre a atividade Dojo de Programação. O primeiro grupo de 22 pessoas era formado por alunos de um curso de especialização da Universidade Tecnológica Federal do Paraná tratado como Grupo 1. O segundo grupo de 11 pessoas era formado por voluntários da comunidade Dojo Paraná tratado como Grupo 2. O Grupo 1 participou do Dojo de Programação compulsoriamente durante uma aula do curso de especialização e o Grupo 2 era composto de pessoas interessadas em Dojo de Programação.

Os questionários foram submetidos aos participantes em datas distintas. O Grupo 1 possuía pouco conhecimento sobre a atividade, vários integrantes do Grupo 2 já haviam participado de

encontros de Dojo de Programação como pode ser visto na Figura 5.

O objetivo desses questionários foi obter as opiniões dos participantes sobre as técnicas usadas no Dojo de Programação, observando as diferenças e semelhanças das respostas de pessoas com maior e menor experiência em participações na atividade e pessoas que participaram compulsoriamente e voluntariamente da atividade.

A Figura 5 apresenta a experiência em números de encontros de Dojo de Programação alegada pelos participantes dos dois grupos. No Grupo 1, a grande maioria, pouco mais de 90% das pessoas nunca participou de encontros de Dojo de Programação. No Grupo 2, pouco mais da metade, quase 55% dos participantes não participou de nenhum encontro.

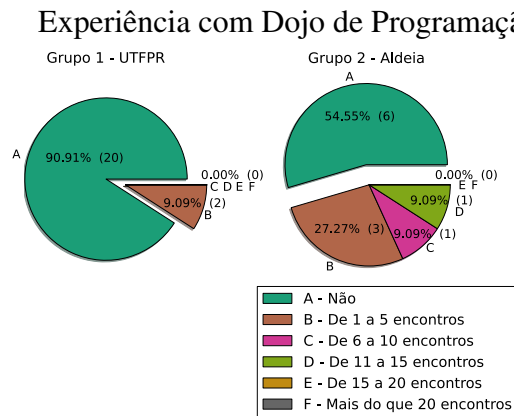


Figura 5: Experiência dos participantes em Dojo de Programação.

A Figura 6 apresentam os resultados da proporção de participantes que afirmaram saber o que é Dojo de Programação. No Grupo 1, menos de 37% dos integrantes afirmaram conhecer o Dojo de Programação. Os participantes do Grupo 2 em sua maioria, pouco menos de 82%, afirmaram conhecer Dojo de Programação, apesar de que alguns afirmaram nunca haver participado.

Conhecimento prévio de Dojo de Programação

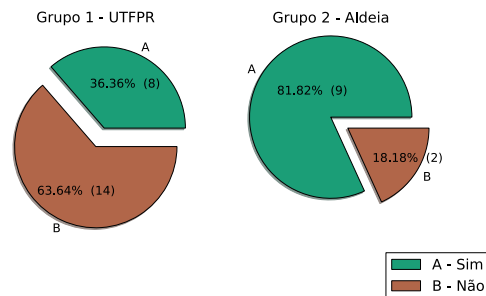


Figura 6: Conhecimento dos participantes sobre Dojo de Programação.

O questionário aplicado após a atividade nos dois grupos coletou informações sobre a opinião dos participantes. As questões foram elaboradas de acordo com os resultados obtidos pelo questionário eletrônico Seção 4.1 e das entrevistas Seção 4.2, Apêndice 7.5. Os resultados do questionário, que denominamos de questionário presencial, permitiram a avaliação da opinião de dois grupos distintos, um grupo de voluntários e um grupo de alunos.

Foram apresentadas questões que usaram a escala de Likert, com cinco opções, Discordo totalmente (-2), Discordo Parcialmente (-1), Neutro (0), Concordo parcialmente (1) e Concordo totalmente (2). As mesmas questões foram apresentadas aos dois grupos. Foi elaborada uma legenda para aprimorar a apresentação dos resultados, como segue:

- Q1 - O dojo de programação não favorece a participação dos alunos.
- Q2 - Comunicação de ideias é um dos pontos fortes do dojo de programação.
- Q3 - Aulas com dojo de programação são muito lentas.
- Q4 - Aulas com dojo de programação são dinâmicas.
- Q5 - Dojos de programação ajudam os programadores a aprender diferentes linguagens.
- Q6 - Há mais interação entre estudantes quando participam de dojos do que quando participam de aulas expositivas.
- Q7 - Aulas com dojo de programação dificultam a concentração.
- Q8 - O dojo de programação ajudou a entender o que é TDD.
- Q9 - É difícil aprender programação em pares com o dojo de programação.

- Q10 - Consegui perceber a importância de passos de bebê durante o dojo de programação.

As questões Q1, Q3, Q7, Q9 tiveram seu sentido invertido para evitar influenciar os participantes a responder sempre de forma positiva. A Tabela 11 contém a distribuição das frequências das respostas em quantidade e percentual do Grupo 1, composto por alunos de curso de especialização da Universidade Tecnológica Federal do Paraná.

Tabela 11: Frequência de respostas do Grupo 1 - UTFPR

	Discordo fortemente		Discordo parcialmente		Neutro		Concordo parcialmente		Concordo totalmente	
	cnt	perc	cnt	perc	cnt	perc	cnt	perc	cnt	perc
Q1	10	47.6%	6	28.6%	1	4.8%	2	9.5%	2	9.5%
Q2	0	0.0%	3	14.3%	1	4.8%	9	42.9%	8	38.1%
Q3	2	9.5%	4	19.0%	4	19.0%	6	28.6%	5	23.8%
Q4	0	0.0%	3	14.3%	1	4.8%	8	38.1%	9	42.9%
Q5	0	0.0%	3	14.3%	0	0.0%	8	38.1%	10	47.6%
Q6	1	4.8%	1	4.8%	2	9.5%	9	42.9%	8	38.1%
Q7	1	4.8%	6	28.6%	2	9.5%	9	42.9%	3	14.3%
Q8	0	0.0%	1	4.8%	3	14.3%	11	52.4%	6	28.6%
Q9	4	19.0%	10	47.6%	1	4.8%	1	4.8%	5	23.8%
Q10	5	23.8%	1	4.8%	5	23.8%	4	19.0%	6	28.6%

As medidas de dispersão, a média, variância e desvio padrão das respostas obtidas de participantes compulsórios do Dojo de Programação realizado em aula do curso de especialização da Universidade Tecnológica Federal do Paraná estão na Tabela 12.

Tabela 12: Média, variância e desvio padrão do questionário sobre o Dojo de Programação da UTFPR - Grupo 1

Variável	Média	Variância	Desvio padrão
Q1	-0.95	1.85	1.33
Q2	1.05	1.05	1.00
Q3	0.38	1.75	1.29
Q4	1.10	1.09	1.02
Q5	1.19	1.06	1.01
Q6	1.05	1.15	1.05
Q7	0.33	1.43	1.17
Q8	1.05	0.65	0.79
Q9	-0.33	2.23	1.46
Q10	0.24	2.39	1.51

Na Figura 7 é possível visualizar a frequência de respostas dos participantes do Dojo de Programação na aula de especialização da Universidade Tecnológica Federal do Paraná representando a tendência das opiniões dos participantes.

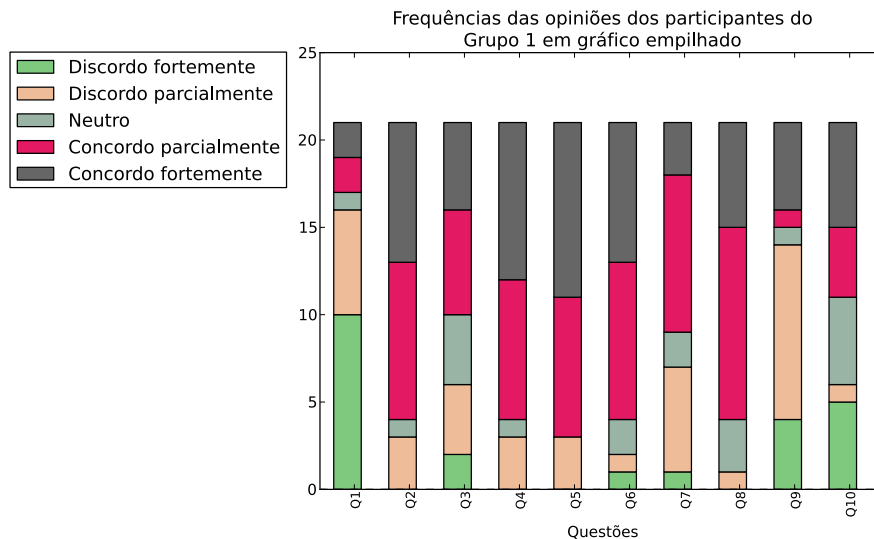


Figura 7: Frequência de respostas dos participantes do Dojo de Programação na UTFPR, Grupo 1

A Tabela 13 contém a distribuição das frequências das respostas em quantidade e percentual do Grupo 2, composto por voluntários membros da comunidade Dojo Paraná, que participaram de um encontro numa sala cedida pela Aldeia CoWorking, um espaço de trabalho colaborativo. Um dos participantes não respondeu a questão 5, a resposta desse participante foi ignorada nos cálculos estatísticos dessa questão.

Tabela 13: Frequência de respostas do Grupo 2 - Aldeia CoWorking

	Discordo fortemente		Discordo parcialmente		Neutro		Concordo parcialmente		Concordo totalmente	
	cnt	perc	cnt	perc	cnt	perc	cnt	perc	cnt	perc
Q1	9	81.8%	2	18.2%	0	0.0%	0	0.0%	0	0.0%
Q2	0	0.0%	0	0.0%	0	0.0%	1	9.1%	10	90.9%
Q3	3	27.3%	2	18.2%	4	36.4%	2	18.2%	0	0.0%
Q4	0	0.0%	0	0.0%	0	0.0%	2	18.2%	9	81.8%
Q5	0	0.0%	0	0.0%	0	0.0%	5	45.5%	5	45.5%
Q6	0	0.0%	0	0.0%	0	0.0%	4	36.4%	7	63.6%
Q7	4	36.4%	5	45.5%	1	9.1%	0	0.0%	1	9.1%
Q8	0	0.0%	0	0.0%	1	9.1%	4	36.4%	6	54.5%
Q9	6	54.5%	5	45.5%	0	0.0%	0	0.0%	0	0.0%
Q10	0	0.0%	1	9.1%	0	0.0%	4	36.4%	6	54.5%

As medidas de dispersão, a média, variância e desvio padrão das respostas obtidas de participantes voluntários do Dojo de Programação realizado em sala cedida pela Aldeia CoWorking estão na Tabela 12.

Tabela 14: Média, variância e desvio padrão do questionário sobre o Dojo de Programação da Aldeia - Grupo 2

Variável	Média	Variância	Desvio padrão
Q1	-1.82	0.16	0.39
Q2	1.91	0.09	0.29
Q3	-0.55	1.27	1.08
Q4	1.82	0.16	0.39
Q5	1.50	0.28	0.50
Q6	1.64	0.25	0.48
Q7	-1.00	1.40	1.13
Q8	1.45	0.47	0.66
Q9	-1.55	0.27	0.50
Q10	1.36	0.85	0.88

Na Figura 8 é possível visualizar a frequência de respostas dos participantes do Dojo de Programação na Aldeia CoWorking representando a tendência das opiniões dos participantes.

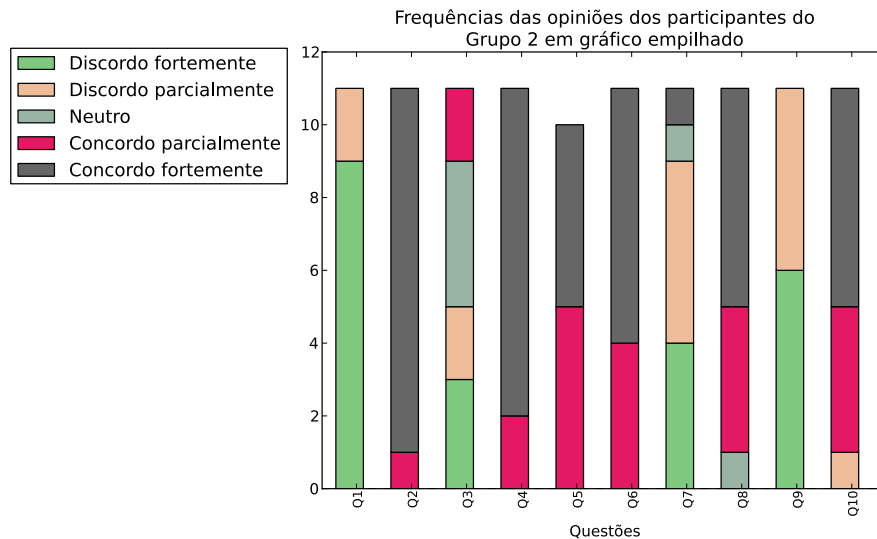


Figura 8: Frequência de respostas dos participantes do Dojo de Programação na Aldeia, Grupo 2

4.4 COMENTÁRIOS SOBRE OS DADOS

O levantamento de dados foi realizado no período de março de 2011 e novembro de 2012. Grupos distintos participaram do levantamento de dados, estudantes de graduação, es-

tudantes de especialização, profissionais que participam de encontros regulares de Dojo de Programação e especialistas com experiência na organização e condução de encontros de Dojo de Programação. O objetivo do levantamento de dados foi identificar a percepção de diversos perfis diferentes de pessoas sobre o ambiente de aprendizado oferecido pelo Dojo de Programação. Foram avaliados aspectos como a interação entre os alunos, a dinâmica e ritmo do encontro bem como práticas ágeis, o desenvolvimento dirigido por o e testes usando dojo de programação, programação em par e passos de bebê, discutidos no capítulo 5.

Para a geração dos gráficos e cálculos estatísticos foram utilizados a linguagem de programação python, a biblioteca matplotlib (HUNTER, 2007) e o editor de imagens vetoriais Inkscape.

5 DISCUSSÃO DOS RESULTADOS

Em um ambiente de ensino, uma sala de aula, seja um curso complementar em uma empresa de treinamento ou um curso universitário, todo o ambiente importa. Os recursos utilizados, a relação dos participantes, o modo como os conceitos são apresentados constroem a experiência de aprendizado.

Dessa forma, o Dojo de Programação foi avaliado tendo em mente a necessidade de observar os mais diversos aspectos que envolvem a experiência do aprendizado. O questionário eletrônico 4.1 foi avaliado comparando respostas de participantes experientes e iniciantes. As entrevistas 4.2 foram avaliadas de forma subjetiva. Os questionários presenciais 4.3 foram comparados entre os participantes compulsórios, Grupo 1 e os participantes voluntários, Grupo 2.

5.1 QUESTIONÁRIO ELETRÔNICO

O questionário eletrônico 4.1 foi dividido em categorias para avaliar diferentes aspectos do ambiente que envolve o Dojo de Programação. Foram avaliados os recursos, a infra-estrutura e as práticas de métodos ágeis utilizadas.

Com relação ao ambiente e recursos do Dojo de Programação foi observado que a maioria dos participantes concordam com as Questões 14, 15 e 19, a maioria discorda da Questão 16 e a maioria também discorda das Questões 17 e 18, que tiveram seus sentidos invertidos. A única questão que apresentou um desvio padrão abaixo de 1 foi a Questão 17 (0.86), sendo assim a questão com a menor dispersão de resposta entre os participantes. Ao avaliar as repostas apenas dos participantes experientes pode ser observado que uma dispersão ainda menor para a Questão 17 (0.29) e uma dispersão relativamente baixa para a Questão 18 (0.72), entretanto o desvio padrão da Questão 14 aumentou (1.31 vs 1.17). Isso indica que os participantes experientes tem uma maior dispersão das respostas para a Questão 14. Ao analisarmos as respostas dos participantes iniciantes percebemos mudanças pequenas no desvio padrão se comparados ao

desvio padrão de todas as respostas. A influencia das respostas dos participantes considerados iniciantes é maior pois são em maior número, 43 dos 64 são considerados iniciantes, 21 são considerados experientes.

Na categoria de perguntas relacionadas aos recursos utilizados, as respostas apresentam as menores médias, com grande dispersão, principalmente as questões 20 e 21. As questões 22 e 23 apresentaram uma média maior, mas ainda com dispersão alta. A análise das respostas de iniciantes e experientes apresenta pouca diferença.

Quanto as perguntas sobre as práticas ágeis usadas no Dojo de Programação, as respostas apresentam uma média alta, com desvio padrão moderado. As duas questões que apresentam um desvio padrão maior foram as questões 25 e 29. As respostas de iniciantes e experientes apresentam alguma diferença, exceto pela questão 25, para todas as respostas apresentou desvio padrão de (1.11), (0.96) para as respostas de experientes e (1.14) para as respostas de iniciantes.

5.2 ENTREVISTAS

As entrevistas realizadas com especialistas apresentaram informações subjetivas sobre a atividade, permitindo uma análise qualitativa sobre o Dojo de Programação. Dos termos comuns extraídos dos discursos e apresentados na Tabela 10 destacam-se a influência do Dojo de Programação na interação entre os alunos, maior participação e trabalho em equipe. Uma consideração se repete, intercalar aulas expositivas para apresentação de conteúdo teórico com a atividade do Dojo de Programação para fixar o conteúdo praticando.

5.3 QUESTIONÁRIO PRESENCIAL

Analisando a dispersão das respostas, o Grupo 1 de participantes compulsórios apresenta altos valores para o desvio padrão 12, o Grupo 2 de participantes voluntários apresenta baixos valores para o desvio padrão 14 na grande maioria das questões. A exceção do Grupo 1 é a questão 8 que apresentou desvio padrão menor que 1 (0.79) e as exceções do Grupo 2 são as questões 3 e 7 que apresentaram desvio padrão pouco maior que 1 (1.08 e 1.13).

Para a discussão dos resultados do questionário presencial os percentuais de concordância parcial e total e os percentuais de discordância parcial e total foram agrupados, apresentados na Tabela 15.

Tabela 15: Comparação de percentual agrupado de respostas do questionário presencial

Questão	Grupo 1 – UTFPR			Grupo 2 – Aldeia		
	Discorda	Neutro	Concorda	Discorda	Neutro	Concorda
Q1	76.2%	4.8%	19.0%	100.0%	0.0%	0.0%
Q2	14.3%	4.8%	81.0%	0.0%	0.0%	100.0%
Q3	28.5%	19.0%	52.4%	45.5%	36.4%	18.2%
Q4	14.3%	4.8%	81.0%	0.0%	0.0%	100.0%
Q5	14.3%	0.0%	85.7%	0.0%	0.0%	91.0%
Q6	9.6%	9.5%	81.0%	0.0%	0.0%	100.0%
Q7	33.4%	9.5%	57.2%	81.9%	9.1%	9.1%
Q8	4.8%	14.3%	81.0%	0.0%	9.1%	90.9%
Q9	66.6%	4.8%	28.6%	100.0%	0.0%	0.0%
Q10	28.6%	23.8%	47.6%	9.1%	0.0%	90.9%

Comparando as respostas dos dois grupos é possível perceber que a maioria das respostas segue a mesma tendência, exceto algumas questões. A maioria dos participantes do Grupo 1 concorda com a questão 3, de que aulas são lentas com Dojo de Programação, enquanto a maioria dos participantes do Grupo 2 discorda. A maioria dos participantes do Grupo 1 concorda com a questão 7, de que o Dojo de Programação dificulta a concentração e a maioria dos participantes do Grupo 2 discordam. Pouco menos da metade dos participantes do Grupo 1 concorda com a questão 10, que o Dojo ajuda a perceber a importância dos passos de bebê, a grande maioria do Grupo 2 também concorda, mas o a dispersão de respostas foi alto no Grupo 1, desvio padrão 1.51 e baixo no Grupo 2, desvio padrão 0.88.

5.4 OBSERVAÇÕES

Os valores de respostas apresentam algumas diferenças quanto ao que se esperava, de acordo com a influência que o Dojo de Programação poderia ter nos participantes. Grande parte das respostas apresentou um desvio padrão relativamente alto, indicando uma variação alta na opinião dos participantes. Os menores valores de desvio padrão foram encontrados nas respostas de participantes voluntários, Grupo 2 do questionário presencial e participantes experientes do questionário eletrônico, indicando que os participantes passam a ter uma percepção diferente da atividade ao longo do tempo. Mesmo com essas diferenças a maioria das questões seguiu a mesma tendência de respostas, em todos os grupos, as exceções foram apontadas anteriormente no Capítulo 4 e nesse Capítulo 5. Os resultados das respostas também acompanharam as opiniões emitidas pelos especialistas nas entrevistas.

6 CONCLUSÃO

Em suma, o Dojo de Programação pode ser utilizado para aprimorar o ensino de programação, especialmente práticas ágeis, melhorando a participação dos alunos e a dinâmica das aulas. Entretanto, deve ser intercalado com aulas expositivas para apresentação de conceitos. Se faz necessário, além disso, utilizar artifícios para ajudar os alunos mais tímidos a perder a inibição inicial. A maioria dos participantes dos questionários alegou que o Dojo de Programação ajudou a aprender as práticas ágeis, programação em par, desenvolvimento dirigido a testes e passos de bebê.

Segundo o relato recorrente dos especialistas, melhorias na participação dos alunos e na dinâmica das aulas foram observados. Ainda, de acordo com os entrevistados, o Dojo é uma atividade prática, ideal para fixar conceitos, mas deve ser intercalado com exposições teóricas para apresentar conceitos. A timidez e motivação dos alunos não foram avaliadas nessa pesquisa, mas nos artigos (CARMO; BRAGANHOLO, 2012) e (DELGADO et al., 2012) os autores afirmaram que alunos relataram sentimento de constrangimento ao participar do Dojo de Programação. As respostas que apresentaram os menores valores de desvio padrão, consequentemente a menor dispersão de opiniões foram as questões relacionadas a práticas ágeis, como foi apresentado na Seção 4.1 e na Seção 4.3. A maioria dos participantes concorda com as afirmações dessa categoria e discorda das afirmações que tiveram seu sentido invertido.

As informações apresentadas nessa pesquisa podem servir como referência para interessados em adotar o Dojo de Programação no ensino de programação, práticas ágeis e até em atividades práticas em geral. Acreditamos ainda que existe uma carência de estudos que avaliem aspectos cognitivos do aprendizado utilizando Dojo de Programação.

Por fim, pudemos perceber um interesse da academia no assunto, tendo em vista a oportunidade de publicar dois artigos em eventos acadêmicos ligados à educação, um artigo (LUZ et al., 2012) no Simpósio Brasileiro de Informática na Educação - SBIE 2013 e um artigo (LUZ et

al., 2013) publicado na *International Conference on Advanced Learning Technologies - ICALT 2013*.

REFERÊNCIAS

- ANICHE, M. F.; SILVEIRA, G. d. A. Increasing learning in an agile environment: lessons learned in an agile team. In: **2011 AGILE conference**. Salt Lake City, UT, USA: [s.n.], 2011. p. 289–295. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6005834>>.
- BACHE, E. Test driven development: performing art. In: ABRAHAMSSON, P. et al. (Ed.). **Agile processes in software engineering and extreme programming**. Springer Berlin Heidelberg, 2009, (Lecture Notes in Business Information Processing, v. 31). p. 217–218. ISBN 978-3-642-01853-4. 10.1007/978-3-642-01853-4_38. Disponível em: <http://dx.doi.org/10.1007/978-3-642-01853-4_38>.
- BALIJEPAALLY, V. et al. Are two heads better than one for software development? the productivity paradox of pair programming. **MIS Q.**, v. 33, p. 91 a 118, mar. 2009. ISSN 0276-7783. Disponível em: <<http://dl.acm.org/citation.cfm?id=2017410.2017418>>.
- BECK, K. **Test driven development: by example**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0321146530.
- BECK, K.; ANDRES, C. **Extreme programming explained: embrace change (2nd edition)**. [S.l.]: Addison-Wesley Professional, 2004. ISBN 0321278658.
- BOSSAVIT, L.; GAILLOT, E. **Coding dojo wiki: BabySteps**. Disponível em: <<http://codingdojo.org/cgi-bin/wiki.pl?BabySteps>>.
- BOSSAVIT, L.; GAILLOT, E. The coder's dojo - a different way to teach and learn programming. In: BAUMEISTER, H.; MARCHESI, M.; HOLCOMBE, M. (Ed.). **Extreme programming and agile processes in software engineering**. Springer Berlin / Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3556). p. 1156–1158. ISBN 978-3-540-26277-0. 10.1007/11499053_54. Disponível em: <http://dx.doi.org/10.1007/11499053_54>.
- BOSSAVIT, L.; GAILLOT, E. **Paris Dojo**. dez. 2005. [Http://codingdojo.org/cgi-bin/wiki.pl?ParisDojo](http://codingdojo.org/cgi-bin/wiki.pl?ParisDojo). Disponível em: <http://codingdojo.org/cgi-bin/wiki.pl?ParisDojo>. Acessado em: 26 de setembro de 2013.
- BRAVO, M.; GOLDMAN, A. Reinforcing the learning of agile practices using coding Dojos. In: SILLITTI, A. et al. (Ed.). **Agile processes in software engineering and extreme programming**. Springer Berlin Heidelberg, 2010, (Lecture Notes in Business Information Processing, v. 48). p. 379–380. ISBN 978-3-642-13054-0. 10.1007/978-3-642-13054-0_41. Disponível em: <http://dx.doi.org/10.1007/978-3-642-13054-0_41>.
- BRAVO, M. V. **Abordagens para o ensino de práticas de programação extrema**. Dissertação (Mestrado) — Instituto de Matemática e Estatística da Universidade de São Paulo, maio 2011. Disponível em: <<http://grenoble.ime.usp.br/gold/orientados/dissertacao-MarianaBravo.pdf>>.

BRUEGGE, B.; REISS, M.; SCHILLER, J. Agile principles in academic education: a case study. In: **2009 sixth international conference on information technology: new generations**. Las Vegas, NV, USA: [s.n.], 2009. p. 1684–1686. Disponível em: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5070901>>.

CARMO, D.; BRAGANHOLO, V. Um estudo sobre o uso didático de dojos de programação. In: **Workshop de educação em computação. sociedade brasileira de computação**. [S.l.: s.n.], 2012.

DELGADO, C.; TOLEDO, R. de; BRAGANHOLO, V. Uso de dojos no ensino superior de computação. In: **Workshop de Educação em Computação (WEI). Sociedade Brasileira de Computação**. [s.n.], 2012. ISSN 2175-2761. Disponível em: <<http://va.mu/dfTz>>.

DOJOPR, G. Wiki, **Home dojo-parana/dojo-parana wiki**. ago. 2012. <https://github.com/dojo-parana/dojo-parana/wiki/>. Disponível em: <<https://github.com/dojo-parana/dojo-parana/wiki/>>.

ERICSSON, K. A.; KRAMPE, R. T.; TESCH-RÖMER, C. The role of deliberate practice in the acquisition of expert performance. **Psychological review**, American Psychological Association, v. 100, n. 3, p. 363, 1993.

GAILLOT, E. **What is Coding Dojo**. 2 set. 2012. Acessado em 13 de setembro de 2013. Disponível em: <<http://codingdojo.org/cgi-bin/wiki.pl?WhatIsCodingDojo>>.

HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing In Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

KOSTER, B. Agile methods fix software engineering course. **J. Comput. Small Coll.**, v. 22, p. 131–137, dez. 2006. ISSN 1937-4771. ACM. Disponível em: <<http://dl.acm.org/citation.cfm?id=1181901.1181925>>.

KVALE, S. **Interviews: an introduction to qualitative research interviewing**. Sage Publications, 1996. ISBN 9780803958197. Disponível em: <<http://books.google.com.br/books?id=tJPZAAAAMAAJ>>.

LEDDY, E. Wiki, **PyClass - noisebridge**. 2012. Disponível em: <<https://www.noisebridge.net/wiki/PyClass>>.

LUZ, R.; NETO, A. G. S. S.; NORONHA, R. V. Usando Dojos de Programação para o Ensino de Desenvolvimento Dirigido por Testes. **Anais do Simpósio Brasileiro de Informática na Educação**. ISSN 2316-6533., nov. 2012.

LUZ, R. B. d.; NETO, A. G. S. S.; NORONHA, R. V. Teaching tdd, the coding dojo style. In: **Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on**. [S.l.: s.n.], 2013. p. 371–375.

MELNIK, G.; MAURER, F. A cross-program investigation of students' perceptions of agile methods. In: **Proceedings of the 27th international conference on software engineering - ICSE '05**. St. Louis, MO, USA: [s.n.], 2005. p. 481. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1062455.1062543>>.

MILLER, J.; SMITH, M. A TDD approach to introducing students to embedded programming. In: **Proceedings of the 12th annual SIGCSE conference on innovation and technology in computer science education - ITiCSE '07**. Dundee, Scotland: [s.n.], 2007. p. 33. Disponível em: <<http://portal.acm.org/citation.cfm?doid=1268784.1268797>>.

PIMENTEL, E. P. et al. Avaliação contínua da aprendizagem, das competências e habilidades em programação de computadores. In: **Anais do Workshop de Informática na Escola**. [S.l.: s.n.], 2003. v. 1, n. 1, p. 533–544.

PRINCE, M. Does active learning work? A review of the research. **JOURNAL OF ENGINEERING EDUCATION**, AMER SOC ENGINEERING EDUCATION, 1818 N ST, N W, STE 600, WASHINGTON, DC 20036 USA, v. 93, n. 3, jul. 2004. Disponível em: <http://www.konferenslund.se/pp/TAPPS_Prince.pdf>.

SATO, D. T.; CORBUCCI, H.; BRAVO, M. V. Coding dojo: an environment for learning and sharing agile practices. **AGILE Conference**, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 459–464, 2008.

SCHNEIDER, J. eXtreme programming - helpful or harmful in educating undergraduates? **Journal of Systems and Software**, v. 74, n. 2, p. 121–132, 2005. ISSN 01641212. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0164121203002929>>.

SINIAALTO, M.; ABRAHAMSSON, P. A comparative case study on the impact of test-driven development on program design and test coverage. In: **Proceedings of the first international symposium on empirical software engineering and measurement**. Washington, DC, USA: IEEE Computer Society, 2007. (ESEM '07), p. 275–284. ISBN 0-7695-2886-4. Disponível em: <<http://dx.doi.org/10.1109/ESEM.2007.2>>.

SKIENA, S. S.; REVILLA, M. A. **Programming challenges: the programming contest training manual**. New York: Springer, 2003. ISBN 0387001638 9780387001630.

THOMAS, D. **CodeKata – How to Become a Better Developer**. 2001. Disponível em: <http://codekata.pragprog.com/>. Acessado em: 18 de abril de 2011. Disponível em: <<http://codekata.pragprog.com/>>.

THOMAS, D. **CodeKata: code kata**. 28 jan. 2007. Disponível em: <http://codekata.pragprog.com/2007/01/code_kata_backg.html>.

THOMAS, D. **CodeKata: code Kata–How it started**. 28 jul. 2007. Disponível em: <http://codekata.pragprog.com/2007/01/code_katahow_it.html>.

VODDE, B.; KOSKELA, L. Learning test-driven development by counting lines. **IEEE Software**, IEEE Computer Society, Los Alamitos, CA, USA, v. 24, p. 74–79, 2007. ISSN 0740-7459.

WILLIAMS, L.; KESSLER, R. **Pair Programming Illuminated**. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0201745763.

7 APÊNDICE: QUESTIONÁRIOS E ROTEIRO DA ENTREVISTA COM ESPECIALISTAS

As questões marcadas com [Escala de Likert] apresentaram as seguintes opções:

- Discordo totalmente
- Discordo parcialmente
- Neutro
- Concordo parcialmente
- Concordo totalmente

7.1 QUESTIONÁRIO ABERTO À COMUNIDADE.

Questionário aprendido com dojo de programação.

Muitos profissionais e empresas têm adotado a atividade dojo de programação para o ensino e a disseminação de práticas ágeis. Os dojos de programação surgiram como uma atividade para o ensino de programação para profissionais atuantes, tendo como público-alvo pessoas que valorizam suas habilidades de programação e possuem uma motivação forte para aperfeiçoar-se. Este questionário faz parte de pesquisa acadêmica na Universidade Tecnológica Federal do Paraná sobre a prática de dojo de programação.

1.Qual a categoria em que você se encaixa?

- Estudante
- Profissional
- Ambos

2.Qual sua área de atuação profissional (cargo/carreira) ou acadêmica (nome do curso)? __

3.Qual sua idade? _____

4.Sabe programar à quanto tempo? _____

5.Gênero:

Masculino

Feminino

6.Você sabe o que é “Test Driven Development”?

Sim

Não

7.Experiência com “Test Driven Development”:

Iniciante

Intermediário

Experiente

8.Você sabe o que é programação em par?

Sim

Não

9.Experiência com programação em par:

Iniciante

Intermediário

Experiente

10.Você sabe o que são passos de bebê (“Baby steps”)?

Sim

Não

11.Experiência com passos de bebê (“Baby steps”):

Iniciante

Intermediário

Experiente

12. Há quanto tempo participa de dojos de programação? _____

13. Já participou de quantos encontros de dojo de programação aproximadamente? _____

14. Com que frequência participa de dojos de programação? *

Semanalmente

Quinzenalmente

Mensalmente

Uma vez a cada bimestre

Uma vez a cada semestre

15. Com relação à atividade de dojo de programação estilo “Randori” (estilo mais usado, não precisa de preparação antecipada, participantes escolhem problema e/ ou linguagem, possui mecanismo para troca de par, começa do zero): [Escala de Likert]

- É melhor quando o intervalo de tempo entre os encontros é menor (por exemplo, semanal é melhor que quinzenal).
- A prática utilizando linguagens de programação conhecida por todos resulta em melhor aprendizado.
- Funciona bem quando todos os participantes são experientes.
- Utilizar a técnica de programação em par atrapalha o andamento da atividade.
- Passos de bebê dificultam o processo de encontrar uma solução para o problema.
- Criar os testes antes do código ajuda o andamento da atividade.

16. Com relação à estrutura física onde ocorre a prática de dojo de programação: [Escala de Likert]

- Laboratórios com computadores são adequados para prática de dojo de programação.
- É importante que o dojo de programação ocorra em um local fixo regularmente.
- É fundamental que exista um quadro ou painel flip-chart para escrever.
- Não é necessário ter um projetor se a dupla comunicar bem a ideia.

17.Com relação à sua percepção do aprendizado durante a atividade de dojo de programação:

[Escala de Likert]

- O dojo de programação favorece o aprendizado de novas linguagens de programação.
- O dojo de programação dificulta o aprendizado de TDD.
- O dojo de programação favorece o nivelamento dos participantes por conta da programação em par.
- Praticar passos de bebê ensina a resolver problemas de forma gradual.
- Durante a atividade de dojo de programação aprende-se bastante devido à troca de experiências.
- Para aprender métodos ágeis prefiro treinamento tradicional, com exposição teórica e exercícios individuais.

18.Comentários: Campo livre para comentários adicionais. _____

- _____
- _____
- _____

7.2 QUESTIONÁRIO PRESENCIAL SOBRE O PERFIL DOS PARTICIPANTES.

Dojo - especialização UTFPR - perfil dos participantes

Questionário parte de pesquisa acadêmica do Mestrado Profissional em Computação Aplicada do Departamento de Informática da Universidade Tecnológica Federal do Paraná.

1.Qual a sua ocupação principal? _____

2.Qual a sua área de atuação? Nome do cargo _____

3.Qual a sua experiência em programação? Anos trabalhados. _____

4.Sabe o que é Desenvolvimento Dirigido por Testes / TDD?

() Sim

Não

5.Qual sua experiência com TDD?

Nenhuma

Menos de 1 ano

Mais de 1 até 2 anos

Mais de 2 até 3 anos

Mais de 3 até 4 anos

Mais de 4 anos

6.Sabe o que é programação em par?

Sim

Não

7.Qual sua experiência com programação em par?

Nenhuma

Menos de 1 ano

Mais de 1 até 2 anos

Mais de 2 até 3 anos

Mais de 3 até 4 anos

Mais de 4 anos

8.Você sabe o que são passos de bebê? Passos de bebê no contexto de desenvolvimento de software.

Sim

Não

9.Qual sua experiência com passos de bebê?

Nenhuma

Menos de 1 ano

Mais de 1 até 2 anos

Mais de 2 até 3 anos

Mais de 3 até 4 anos

Mais de 4 anos

10. Conhece dojo de programação?

Sim

Não

11. Já participou de encontros de dojo de programação?

Não

De 1 a 5 encontros

De 6 a 10 encontros

De 11 a 15 encontros

De 15 a 20 encontros

Mais do que 20 encontros

7.3 QUESTIONÁRIO PRESENCIAL SOBRE A OPINIÃO DOS PARTICIPANTES.

Dojo - Especialização UTFPR - opinião dos participantes

Questionário parte de pesquisa acadêmica do Mestrado Profissional em Computação Aplicada do Departamento de Informática da Universidade Tecnológica Federal do Paraná.

1. Responda as questões de acordo com a sua opinião quanto à sua participação no dojo de programação. [Escala de Likert]

- O dojo de programação não favorece a participação dos alunos.
- Comunicação de ideias é um dos pontos fortes do dojo de programação.
- Aulas com dojo de programação são muito lentas.
- Aulas com dojo de programação são dinâmicas.
- Dojos de programação ajudam os programadores a aprender diferentes linguagens.
- Há mais interação entre estudantes quando participam de dojos do que quando participam de aulas expositivas.
- Aulas com dojo de programação dificultam a concentração.

- O dojo de programação ajudou a entender o que é TDD.
- É difícil aprender programação em pares com o dojo de programação.
- Consegui perceber a importância de passos de bebê durante o dojo de programação.

7.4 QUESTÕES DAS ENTREVISTAS

As entrevistas seguiram um roteiro semi-estruturado, com liberdade para o entrevistado expressar sua opinião.

- Nome.
- Profissão.
- Como conheceu Dojo de Programação?
- Qual sua experiência com Dojo de Programação?
- Na sua opinião, quais as vantagens do Dojo de Programação em relação a aulas expositivas?
- Na sua opinião, quais as desvantagens do Dojo de Programação em relação a aulas expositivas?
- Quais práticas ágeis têm o entendimento facilitado ao serem ensinadas com Dojo de Programação?
- Dentre essas práticas alguma em especial é preferível ser ensinada com Dojo de Programação?
- Você acha que alguma prática ágil não deve ser ensinada com Dojo de Programação?
- Você tem algum comentário adicional?

7.5 RESUMO DAS ENTREVISTAS COM ESPECIALISTAS EM DOJO DE PROGRAMAÇÃO

Especialistas com larga experiência em Dojo de Programação foram entrevistados. Os especialistas escolhidos publicaram artigos acadêmicos sobre Dojo de Programação ou foram indicados diretamente por especialista que publicou artigo acadêmico sobre Dojo de Programação. O resumo das entrevistas é apresentada à seguir. As entrevistas estão organizadas em ordem alfabética.

7.5.1 Alexandre Freire

Alexandre Freire é mestre em Ciência da Computação pelo Instituto de Matemática e Estatística - IME (USP). Trabalha com desenvolvimento de sistemas de software e consultoria e treinamento em métodos ágeis. É sócio fundador da Agilbits, empresa de consultoria e desenvolvimento.

Conheceu Dojo de Programação em 2005 na conferência de programação extrema - XP, realizada na Universidade de Sheffield, Inglaterra, durante o workshop apresentado por Laurent Bossavit e Emmanuel Gaillot The Coder's Dojo. Ao retornar ao Brasil, em companhia de Danilo Sato, Hugo Corbucci e Mariana Bravo, começou a realizar encontros no IME - USP.

Durante a entrevista destacou algumas características da atividade: a repetição de exercícios, a motivação e o aprendizado de novas linguagens. Além de participar dos encontros no IME-USP participou de encontros na empresa onde trabalhou.

Afirmou que a atividade trouxe crescimento pessoal através da prática contínua. Sobre os passos de bebê, comentou que após algum tempo essa técnica, pode ser bem compreendida pelo aprendiz de forma a permitir que se abra mão do seu uso no dia-a-dia profissional em alguns casos.

Alexandre, quando questionado sobre sua opinião se o Dojo de Programação oferece alguma vantagem se comparado a aulas expositivas, afirmou que uma das grandes vantagens é a participação, favorecendo a comunicação e troca de ideias em uma atividade participativa.

Quanto à sua opinião sobre a desvantagem de Dojo de Programação em relação à uma aula expositiva comentou que considera o Dojo de Programação interessante por focar em aspectos específicos, em contrapartida aulas expositivas podem dar uma visão abrangente sem restringir como o Dojo de Programação.

Em aulas expositivas o foco é na teoria, na apresentação de conceitos e no Dojo de Programação há um grande foco na prática. Segundo Alexandre Freire o Dojo de Programação foi concebido para a prática de desenvolvimento dirigido por testes (TDD), mencionou ainda que o Dojo ensina a fazer retrospectiva que faz parte do processo de melhoria contínua. Esta

atividade é sempre realizada nos minutos finais do encontro onde os participantes expõem suas opiniões sobre os pontos positivos e negativos do encontro.

Perguntado sobre para que tipo de ensino o Dojo não é indicado, Alexandre sugeriu que Dojo de Programação não é indicado para aprender uma linguagem que os participantes não conhecem; é mais produtivo ter ao menos um especialista da linguagem para que a atividade traga resultados úteis.

Em suas considerações finais Alexandre comparou Dojos de Programação à práticas de atividades físicas, sugerindo que o incentivo de um grupo de amigos ajuda a motivar, tanto a prática de atividades físicas como a prática de programação.

7.5.2 Daniel Cukier

Daniel Cukier é diretor de tecnologia da Elo7, um market place de artesanato, um mercado onde artesões podem vender seus produtos. Conheceu o Dojo de Programação no IME-USP, com Danilo Sato e Hugo Corbucci. Começou então a coordenar um grupo na empresa onde trabalhava, Locaweb.

De acordo com Daniel, o início foi um pouco difícil pois as pessoas achavam que Dojo de Programação não fazia parte do trabalho, as pessoas tinham receio de participar por receio de serem mal interpretadas pelos superiores e seus colegas de time.

Quando tentaram uma alternativa, fazer fora do horário de trabalho, passaram por outra dificuldade, as pessoas já haviam ido embora e não apareciam para participar. Aos poucos a atividade foi ganhando adeptos e apoio das diretorias, que através de mensagens de e-mail convidavam e incentivavam os funcionários a participar.

Em seguida os encontros foram abertos para comunidade, isto é pessoas que não trabalhavam na Locaweb puderam participar. Nesse momento o Dojo de Programação começou a fazer mais sucesso, entretanto criou uma dependência de uma pessoa que organizasse o encontro, enviando convites e tomando a frente.

Segundo Daniel, os encontros pararam de ocorrer quando ele parou de organizar. Ele afirmou que para o sucesso do Dojo de Programação é necessário existir essa pessoa que abrace a ideia e faça acontecer.

Questionado sobre os resultados dos encontros comparou o grupo do IME-USP com o grupo da Locaweb. No IME-USP o grupo se reúne regularmente e grande parte dos membros são fixos, assim puderam aprofundar conceitos gradualmente.

No grupo da Locaweb, pelas características particulares, o tamanho da empresa e pelo andamento dos encontros, novos membros compareciam, sendo então necessário iniciar o Dojo de Programação de um nível básico, introdutório.

Relatou ainda que, mesmo com essa particularidade, após cerca de 20 encontros, os participantes puderam entender na prática o que é TDD. Afirmou que o Dojo de Programação tem outro aspecto marcante, socializar o desenvolvimento, fortaleceu a cultura da programação em par e ajudou a humanizar a atividade de programação.

Comentou sobre variações do Dojo de Programação: Dojo para geração de imagens onde o teste não é automatizado mas visual, Dojo para máquinas virtuais com teste de serviços e o formato Kake onde vários pares programam em paralelo com problemas e/ou linguagens diferentes.

Adicionalmente sugeriu que o Dojo de Programação pode ser uma boa ferramenta de contratação. Na Locaweb alguns participantes dos encontros de Dojo de Programação foram contratados.

Segundo Daniel existem duas coisas importantes no Dojo de Programação: a retrospectiva que permite a reflexão sobre a atividade e a comida, que permite uma confraternização e troca de informações mais espontânea entre os participantes.

Finalizou comentando que o Dojo estimula o trabalho em equipe e a interação entre pessoas, duas características importantes nas empresas.

7.5.3 Daniel Wildt

Daniel Wildt trabalha com desenvolvimento de software há mais de 15 anos. Atuou como professor universitário por mais de 7 anos. Começou a usar o Dojo de Programação no ambiente acadêmico em 2005. Durante as aulas de programação em Java reparou que os alunos não conseguiam acompanhar o conteúdo da disciplina.

Utilizou inicialmente a programação em par, entretanto, aconteceu de alguns pares estarem mais engajados do que outros. Começou então a trabalhar com uma máquina e um par apenas, o restante dos alunos permaneciam ao redor da máquina e a cada 5 minutos havia uma troca da dupla.

Em suas aulas a primeira parte era expositiva com alguma teoria e a segunda parte usava a atividade do Dojo de Programação. Usavam diferentes formas de Dojo de Programação: *Prepared, Randori e Kake*.

Segundo Daniel Wildt, uma vantagem dessa atividade é a utilização, desde a primeira aula, de testes automatizados ao invés de usar a saída do console para testar o programa. Daniel Wildt então percebeu um maior engajamento dos alunos, que começaram a participar mais da aula e puderam mostrar suas qualidades diariamente. Alguns receberam indicação para oportunidades profissionais.

Questionado sobre a continuidade do Dojo de Programação, afirmou que após sua saída da faculdade um aluno continuou, entretanto a participação diminuiu e na data da entrevista o Dojo de Programação na faculdade não ocorria mais.

De acordo com Daniel Wildt, o Dojo de Programação facilita o entendimento de algumas práticas, técnicas e tecnologias, dentre elas o *design* simples, ritmo de trabalho, automação de testes, orientação a objetos e padrões de projeto, introdução à novas tecnologias como a linguagem Ruby, linguagem Javascript.

Outro ponto citado por Daniel Wildt foi a mistura de tecnologias. Em uma disciplina de Java realizou o Dojo de Programação no formato *Kake*, com três computadores e três duplas, o

mesmo problema e três linguagens diferentes, incitando a curiosidade e apresentado a diversidade de opções que profissionais tem à disposição.

7.5.4 Danilo Sato

Danilo Sato é atualmente consultor da Thoughtworks. Conheceu Dojo de Programação em 2006, durante um workshop organizado por Emilie Bachè durante o evento Agile Processes in Software Engineering and Extreme Programming. Ao voltar ao Brasil deu início aos encontros no IME-USP.

Segundo Danilo, o Dojo de Programação é mais interativo que aulas expositivas. Durante o Dojo de Programação todos resolvem o exercício ao mesmo tempo. De acordo com Danilo, a experiência de construir a solução em cima do que outros fizeram ajuda a fixar o conteúdo.

Citou TDD e programação em par como práticas ágeis para as quais o Dojo de Programação favorece o ensino. Danilo comentou que eventualmente, quando não pode participar de encontros de Dojo de Programação, utiliza algumas técnicas do Dojo de Programação sozinho. Obviamente não faz programação em par, mas utiliza TDD e baby steps para resolver desafios enquanto passa o tempo durante alguma viagem de avião.

7.5.5 Elizabeth Leddy

Elizabeth Leddy, consultora de software, integrante do time de desenvolvimento do Plone, sistema de gerenciamento de conteúdo livre, concedeu uma entrevista não estruturada. O autor achou interessante entrevistá-la devido à experiência realizando Dojos de programação em suas aulas.

Elizabeth é voluntária num espaço hacker mais conhecidos pelo termo em inglês *hackerspace*, chamado Noisebridge em São Francisco, Califórnia, onde ensina programação com a linguagem Python (LEDDY, 2012) para pessoas que querem mudar de carreira e desenvolver software para internet. As aulas já estão acontecendo desde 2011. Na data da entrevista, em 2012, já haviam sido realizadas cerca de três turmas, aproximadamente 50% dos participantes estavam desde o começo e os outros 50% variam a cada nova turma.

Elizabeth relatou que, no começo, durante as aulas alguns alunos entendiam rapidamente o assunto e alguns não entendiam, mas ao invés de tirarem as dúvidas eles ficavam quietos.

Após conhecer o Dojo de Programação ela experimentou usar durante as aulas, comentou que a atividade se tornou extremamente popular rapidamente. O tópico ensinado é particularmente complexo, ensinar desenvolvimento web para pessoas que não sabem programar e que querem mudar de carreira. Esse perfil de pessoas precisa aprender conceitos práticos para que possam aplicar em projetos pessoais, num ritmo razoável, sem pressa.

O Dojo de Programação ajudou-os com isso pois a aula segue no ritmo da pessoa que está usando o computador, chamado de piloto. Outro ponto relevante é que as pessoas interagem mais, perguntam mais coisas uns aos outros e criam laços que se estendem para o dia a dia após as aulas. Alguns alunos começaram projetos pessoais juntos.

A estrutura das aulas é semanal e organizada de forma que a cada semana seja intercalada uma aula para apresentar conceitos e uma aula prática com Dojo de Programação.

Segundo Elizabeth, o Dojo de Programação torna a aula mais dinâmica. Eventualmente, quando a dupla que está usando o computador não consegue resolver alguma coisa, é convidado um aluno que tem um pouco mais de conhecimento e o que acontece é que esse aluno, pela confiança e segurança no assunto, fala bastante e ajuda o grupo a caminhar, tornando o ambiente participativo.

7.5.6 Hugo Corbucci

Hugo Corbucci é programador e consultor, atualmente trabalhando na Thoughtworks. Conheceu o Dojo de Programação com o Danilo Sato em 2006, que voltava da conferência Agile Processes in Software Engineering and Extreme Programming e organizou o primeiro encontro no Instituto de Matemática e Estatística da Universidade de São Paulo - IME-USP.

Continuou participando de encontros no IME-USP, participou de outros encontros: em Grenoble e Paris na França e alguns nos Estados Unidos. Tiveram um artigo aceito na Agile Conference em 2008 (SATO et al., 2008).

Segundo Hugo, o Dojo de Programação oferece um ambiente colaborativo favorecendo a participação permitindo assim que cada participante aproveite melhor a experiência do aprendizado.

Questionado sobre as desvantagens do Dojo de Programação, afirmou que existem momentos em que é necessário ficar em silêncio, sozinho e refletir sem a pressão do tempo, usado no Dojo de Programação para troca de pares e sem a pressão da plateia observando. Isto permite ao aluno absorver melhor alguns tipos de conteúdo.

O Dojo de Programação é mais indicado para prática. Como práticas ágeis que o Dojo de Programação facilita o aprendizado citou: programação em par, testes de unidade, TDD após o aprendizado de testes de unidade e desenvolvimento incremental. Este último, facilitado pelo Dojo de Programação, ajuda os participantes a perceberem que é possível dar um passo de cada vez sem ter que resolver o problema todo de uma só vez.

Finalizou comentando que percebeu um crescimento do interesse por Dojo de Programação. A lista de discussão do Dojo São Paulo possui muitos participantes e há um interesse crescente das pessoas em praticar e continuar aprendendo.

7.5.7 Mauricio Aniche

Mauricio Aniche é consultor e instrutor na Caelum. Conheceu Dojo de Programação no IME-USP. Participou de encontros na Locaweb, na Caelum e em outros lugares.

Segundo ele, Dojo de Programação é uma ferramenta que ajuda os participantes a perderem a inibição, uma prática que ajuda a socialização dos participantes.

Questionado sobre a comparação entre aulas expositivas e Dojo de Programação afirmou que o Dojo de Programação é bastante prático. Apesar de não ser um substituto para aulas expositivas pode ser usado em conjunto.

Mauricio relatou que o Dojo de Programação precisa ser guiado para ser útil. A ajuda de alguém especialista no assunto ou na linguagem de programação escolhida para o Dojo de Programação favorece o aprendizado.

Para Mauricio, o Dojo de Programação é especialmente indicado para ensinar programação em pares. Se for bem guiado é interessante para ensinar TDD e práticas de design e usar as ferramentas de integração contínua assim como a disciplina necessária para integração contínua.

Relatou ainda sua experiência na Caelum, onde Dojo de Programação foi utilizado pela equipe de instrutores. Após algum tempo a equipe perdeu o interesse em Dojo de Programação pois não chegavam ao fim da solução de problemas relativamente simples. Decidiram substituir o Dojo de Programação por palestras técnicas. Experimentou usar Dojo de Programação em um treinamento, mas ainda não chegou a avaliar o resultado dessa experiência.

Acredita que o Dojo de Programação pode ser usado para ensinar programação tanto em universidades quanto em centros de treinamento. Comentou que acredita que o Dojo de Programação possa ser aprimorado se possuir objetivos claros e diretrizes.

7.5.8 Roberto Rodrigues

Roberto Rodrigues é aluno da Escola de Artes, Ciências e Humanidades da Universidade de São Paulo (EACH-USP) e trabalha no Centro de Competência em Software Livre da Universidade de São Paulo (CCSL-USP). Pesquisa usabilidade e metodologias ágeis na universidade.

Conheceu o Dojo de Programação no IME-USP. Organizador dos Dojos de Programação na EACH-USP. Na entrevista destacou o aspecto colaborativo do Dojo de Programação. Segundo ele, mesmo pessoas experientes tem muito a aprender durante os encontros de Dojo de Programação.

Ao ser questionado sobre a relação entre Dojo de Programação e aulas expositivas, afirmou que o Dojo de Programação favorece o aprendizado na prática pois é comum o aluno acreditar que entendeu a teoria e no momento em que vai tentar fazer sozinho perceber que não entendeu. No Dojo de Programação ele tem a chance de tirar a dúvida imediatamente já que todos estão observando e dispostos a ajudar, enquanto que nas aulas expositivas essa dúvida nem sempre é levada ao professor ou instrutor imediatamente.

Relatou que ainda não conseguiu bons resultados com alunos que não sabiam nada e aprenderam a programar exclusivamente com Dojo de Programação, mas afirmou que a curva de aprendizado para alunos que já sabiam o básico de programação é muito rápida.

Uma desvantagem do Dojo de Programação, segundo Roberto, é que foca muito na prática, perdendo assim um pouco do conceito, sendo necessário usar de artifícios adicionais para resolver isso, combinando alguma atividade para cobrir a parte teórica que falta ao Dojo de Programação.

Segundo ele, o Dojo de Programação é especialmente indicado para ensinar as práticas que exigem disciplina ou são muito repetitivas, como TDD, integração contínua e retrospectiva.

Finalizou comentando que o Dojo de Programação precisa de uma introdução teórica sobre as técnicas usadas para melhorar os resultados da atividade, ele acha essencial que o instrutor deixe claro a importância de tirar as dúvidas durante a atividade.

7.5.9 Considerações sobre as entrevistas

Durante as entrevistas foi possível perceber que os especialistas compartilham de algumas opiniões semelhantes. O foco em atividades práticas, intercalar a atividade com apresentações teóricas, a maior interação entre os alunos, são ponto passivo entre os especialistas.

Entretanto algumas características dividem opiniões. Um dos especialistas afirma que experimentar sozinho sem a pressão do tempo e da plateia observando ajuda a fixar alguns conceitos. Outro especialista afirma que o Dojo de Programação ajuda o aluno a perder a inibição. Enquanto um especialista afirma que o Dojo não é indicado para aprender uma linguagem nova, outro afirma que o ambiente do Dojo de Programação é ideal para aprender linguagens novas.

Comparando essas opiniões com artigos (DELGADO et al., 2012) e (CARMO; BRAGANHOLA, 2012), verificamos que foram observados comportamentos semelhantes. Nos artigos foi relatada a timidez, insegurança e a falta de interesse em participar, sendo necessário oferecer incentivos. Nos artigos não foi mencionado o foco em atividades práticas.