

# Processo de Desenvolvimento de Software: uma Atividade Prática Supervisionada baseada nos papéis de adquirente e fornecedor

André Roberto Ortoncelli<sup>1</sup>, Marisângela Pacheco Brittes<sup>1</sup>

<sup>1</sup>Universidade Tecnológica Federal do Paraná - (UTFPR) Câmpus Dois Vizinhos  
Estrada p/ Boa Esperança, S/n - Zona Rural, Dois Vizinhos - PR, 85660-000

**Abstract.** *This paper presents an experience report concerning a work applied as Activity Supervised Practice, in a Software Development Process discipline, offered in a Software Engineering undergraduate course. In this activity the students were divided in groups, each group at different times represented the software acquirer and supplier roles (such as is defined in ISO/IEC 12207:2008), and the interaction between different groups it was compulsory. During the development of the activity, the groups had to follow some of the MPS.BR requirements, thus reinforcing concepts of software process quality learned in class. It is noteworthy that according to the opinion of students (obtained through a questionnaire), this work contributed to acquisition and fixation of the knowledge about the discipline, as well as to keep them motivated during its development.*

**Resumo.** *Esse trabalho apresenta um relato de experiência referente a um trabalho aplicado como Atividade Prática Supervisionada, em uma disciplina de Processo de Desenvolvimento de Software, ofertada em um curso de graduação em Engenharia de Software. Nessa atividade os alunos foram divididos em grupos, sendo que cada grupo, em diferentes momentos representou os papéis de adquirente e fornecedor de um software (tal como é definido na norma ISO/IEC 12207:2008), sendo obrigatória a interação entre diferentes grupos. Durante o desenvolvimento da atividade os grupos tiveram que seguir alguns dos requisitos do MPS.BR, reforçando assim conceitos de qualidade de processo de software aprendidos em aula. Destaca-se que de acordo com a opinião dos alunos (obtida por meio de um questionário), esse trabalho contribuiu para aquisição e fixação dos conhecimentos da disciplina, assim como para mantê-los motivados durante seu desenvolvimento.*

## 1. Introdução

Cursos de graduação em Engenharia de Software são relativamente novos no Brasil. Tal curso, envolve conhecimentos sobre desenvolvimento, implantação e gerenciamento de projetos de software e também conhecimento para uso de processos organizacionais, passando pela infraestrutura e manutenção de um software, visando um perfil de egresso que deve dominar técnicas de desenvolvimento e aprimoramento de processos e produtos de software.

A disciplina de Processo de Desenvolvimento de Software (PDS) envolve conceitos de produto, processo e projeto de software, apresentando diferentes processos

de desenvolvimento (ágeis e tradicionais), e também modelos que qualidade de processo tal como MPS.BR e CMMi. Trata-se de uma disciplina que dificilmente é encontrada em cursos de graduação na área da computação, apesar de seus conteúdos não serem novos, eles comumente estão divididos em outras disciplinas. De acordo com [Kuhrmann et al. 2013], apesar de processos de desenvolvimento de software serem uma parte essencial da Engenharia de Software, eles são apenas ligeiramente incluídos em currículos de cursos de desenvolvimento de software típicos.

Os conceitos relacionados a PDS são de extrema importância na formação do futuro engenheiro de software, devendo ser considerado alguns requisitos básicos para que seu processo de ensino e aprendizagem seja eficaz, tal como os apresentados em [Kuhrmann et al. 2013], que define cinco requisitos para um curso de PDS : (i) Os alunos precisam compreender a importância de modelos de processo; (ii) Os alunos precisam compreender a integração dos modelos de processos em um ambiente organizacional; (iii) Os alunos precisam experimentar modelos de processos realistas em termos de diferenças de tamanho e complexidade relacionados com as pequenas, médias e grandes organizações e projetos; (iv) Os alunos precisam de conhecimento diferenciado sobre as diferentes abordagens de modelagem; e (v) Os alunos precisam aprender os modelos de processos mais relevantes existentes (tais como padrões nacionais ou internacionais).

Além dos requisitos apresentados em [Kuhrmann et al. 2013], a disciplina de PDS também enfrenta uma série de desafios comuns ao processo de ensino e aprendizagem de Engenharia de Software [Paiva 2011]: (i) distância entre o conteúdo ensinado e a realidade do mercado de trabalho; (ii) falta de interesse do discente em disciplinas maciçamente teóricas; (iii) altas taxas de evasão e reprovação; (iv) índices de aprendizagem efetiva bastante questionáveis; (v) pouca ênfase no trabalho em grupo; (vi) pouco estímulo à interdisciplinaridade; (vii) o planejamento de conteúdos e práticas pedagógicas são ditados pelo interesse de cada docente sem passar, muitas vezes, pelo crivo institucional; (viii) de forma geral, o que se ensina e aprende em um semestre ou disciplina, não é apropriado pela comunidade como um patrimônio intelectual para uso futuro; (ix) pouca ou nenhuma interação entre professores e alunos fora da sala de aula; (x) o conhecimento produzido em um curso/disciplina não é compartilhado com a comunidade externa; e (ix) cada instituição / docente segue sua própria linha de acordo com a sua visão/concepção.

Portanto, ministrar e planejar uma disciplina de PDS é uma tarefa que conta com uma série de desafios, além dos desafios apresentados em [Kuhrmann et al. 2013] e [Paiva 2011], outro desafio importante é não abordar de forma aprofundada conteúdos que são trabalhados com maior ênfase em outras disciplinas, e da mesma forma não se pode abordar de forma superficial conteúdos que serão trabalhados com menor intensidade em outras matérias, tal como a disciplina de Gerência de Projetos, que pode abordar a gestão no contexto de metodologias ágeis, e a disciplina de Qualidade de Software que irá abordar conceitos de qualidade de processo, entre outras matérias de um curso de Engenharia de Software.

A disciplina de PDS contemplada nesse relato de experiência foi ministrada no segundo semestre de um curso de Engenharia Software, sendo assim os alunos ainda não haviam aprendido todos os conceitos relacionados a processos de desenvolvimento de software (por exemplo, os alunos ainda não haviam tido aulas de teste de software, análise e projeto orientado a objetos e gerência de configuração), e não há tempo para se

aprofundar devidamente nesses conteúdos durante a disciplina de PDS. Nesse contexto, destaca-se o desafio de manter os alunos interessados e motivados em relação as aulas da disciplina e também no desenvolvimento da Atividade Prática Supervisionada (APS).

Uma APS consiste em uma ou mais atividades extraclasse, que devem ser acompanhadas pelo professor da disciplina, pois integralizam a carga horária da disciplina, podendo compor também a nota do aluno na disciplina.

Nesse contexto, o objetivo desse trabalho é apresentar um relato de experiência referente a aplicação de uma APS, na disciplina de PDS, baseada na interação entre diferentes grupos de alunos, que em diferentes momentos trabalham como adquirente e fornecedor de um processo de desenvolvimento de software (permitindo assim simular um processo de desenvolvimento de software real), de acordo com a norma NBR ISO/IEC 12207:2008 [ISO 2008], tendo que lidar com requisitos de qualidade do MPS.BR [SOFTEX 2012].

O restante desse trabalho encontra-se organizado da seguinte maneira: A seção 2 apresenta uma breve revisão de literatura referente ao tema. A proposta metodológica da APS que é foco desse relato de experiência é apresentada na seção 3. O questionário de avaliação utilizado e resultados obtidos com sua aplicação serão apresentados na seção 4, sendo uma análise de tais resultados apresentada na seção 5. Por fim, a seção 6, apresenta uma breve conclusão.

## **2. Revisão de Literatura**

Tradicionalmente o ensino de Engenharia de Software se dá por meio de aulas teóricas e aplicação de exercícios práticos. Porém recentemente vêm sendo introduzidas alternativas lúdicas nesse processo, tais como jogos e simuladores. Nesse contexto em [Kohwalter et al. 2014] é apresentada uma análise positiva em relação a como tais técnicas podem levar a respostas mais rápidas e precisas dos estudantes, incluindo os aspectos de aprendizagem que não poderiam ser alcançados da forma tradicional.

Com relação á Engenharia de Software, de forma geral, pode-se destacar vários trabalhos recentes que tem como objetivo tornar o processo de ensino e aprendizagem mais efetivo e motivador. No entanto, quando se foca apenas na área de PDS, o número de trabalhos é ainda um pouco limitado. O objetivo dessa seção é listar alguns desses poucos trabalhos existentes na literatura.

Em [Prikladnicki et al. 2009] são apresentadas diferentes estratégias para o ensino de Engenharia de Software, sendo uma delas focada no uso de LEGO para o ensino de processo de desenvolvimento de software, abrangendo Normas ISO, processos ágeis e de forma geral os modelos de qualidade CMMI e MPS.BR, objetivando consolidar os conceitos dos processos de gerência de projetos, gerência de requisitos, garantia da qualidade, medição e gerência de configuração do MPS.BR. Na atividade apresentada, um produto deve ser construído a partir dos requisitos de forma planejada, monitorada e controlada, de forma similar a um processo de desenvolvimento de software, porém com uso de LEGO.

Técnicas de gamificação são exploradas na abordagem proposta em [Cavalcante et al. 2015], que conseguiu um aumento de 40% na aderência aos processos organizacionais e de 214% na capacitação dos colaboradores. Nesse trabalho se

desenvolveu uma competição lúdica envolvendo todos os projetos de uma organização, com o objetivo de melhorar alguns aspectos relacionados a qualidade do produto. Para pontuação no jogo foram considerados itens em comum a todos os projetos da organização, tais como aderência aos processos organizacionais e capacitação dos colaboradores nos processos.

Em [Werneck et al. 2012] é apresentada uma abordagem interdisciplinar para o ensino de Engenharia de Software para alunos de três disciplinas distintas. Nessa abordagem, a partir da definição de um processo formal centrado nas atividades de gerência de projetos, os alunos da disciplina de Projeto de Sistemas de Informação são responsáveis por desenvolver um sistema com base em uma especificação de requisitos recebida no início do semestre, seu trabalho é gerenciado por um grupo de alunos da disciplina de Gerência de Projetos, e alunos da disciplina de Engenharia de Software II realizam a análise de pontos de função e atuam também em atividades relacionadas a qualidade de software, tal como, inspeção de código.

Outra, atividade interdisciplinar foi aplicada por [Aguilar et al. 2015], na disciplina de Engenharia de Software Aplicada, foi solicitado que os alunos utilizassem um processo de desenvolvimento de software para produção de um software para resolver um problema não fictício, e na disciplina de Gestão da Qualidade de Software os alunos foram guiados na produção e aplicação de um plano de avaliação de qualidade, que avaliou entre outros itens a qualidade do processo de desenvolvimento utilizado e do software codificado.

Em [Bavota et al. 2012] também é apresentada uma atividade interdisciplinar que visa explorar o cenário real de desenvolvimento de software. Alunos da disciplina de Engenharia de Software (de curso de graduação) e alunos da disciplina de Gerenciamento de Projeto de Software (de um programa de mestrado), tiveram que interagir entre si, sendo que os alunos de graduação tiveram funções de desenvolvimento, enquanto os alunos de mestrado foram os responsáveis pelo gerenciamento, visando simular um cenário real, no qual a gestão de qualidade foi um dos conceitos chave.

Além dos trabalhos citados, que tem uma ênfase maior em processos tradicionais, existem na literatura alguns trabalhos focados no ensino de processos ágeis de desenvolvimento de software [Gestal and Barros 2014, Rodriguez et al. 2015].

Em [Rodriguez et al. 2015] é apresentado um ambiente virtual intitulado Virtual Scrum, que utiliza recursos tais como quadros, gráficos e calendários, para simular a sala de uma equipe Scrum. Além disso também permite que avatares 3D interajam uns com os outros. Simulando de forma realista um processo de desenvolvimento ágil com a metodologia Scrum.

Também focado em metodologias ágeis, em [Gestal and Barros 2014] é apresentado um modelo de simulação (e também um criador de simulações) de curta duração (duas a quatro horas) para o ensino da metodologia de desenvolvimento Scrum, com ênfase na tomada de decisões e feedback.

Outro trabalho que simula um processo de desenvolvimento de software por meio de um jogo é apresentado em [Monsalve et al. 2015], nesse jogo o aluno pode assumir diferentes papéis durante o processo de desenvolvimento de um software, lidando com vários problemas recorrentes do desenvolvimento, ao explorar conceitos de complexi-

dade, tamanho e qualidade do produto, de produtividade e maturidade da equipe de desenvolvimento, de gestão de orçamento, de riscos do desenvolvimento, qualidade e aceitação do produto final. Além disso, tal jogo também pode ser jogado de forma individual e coletiva, sendo dividido em rodadas nas quais, os jogadores executam uma série de ações, participando de forma ordenada.

### 3. Procedimento Metodológico

A APS descrita nesse texto foi aplicada no segundo semestre do ano de 2015, na disciplina de PDS para uma turma de 29 alunos, e está sendo aplicada novamente no primeiro semestre de 2016, para uma turma de 32 alunos.

Nessa APS, os alunos foram divididos em grupos de 3 ou 4 integrantes. Tais grupos interagiram entre si, pois em diferentes momentos representaram os papéis de adquirente e fornecedor em um processo de desenvolvimento de software, de acordo com a norma ISO/IEC 12207:2008 [ISO 2008].

Sendo  $S = \{s_1, s_2, \dots, s_x\}$  o conjunto dos  $x$  temas de software propostos, e  $G = \{g_1, g_2, \dots, g_n\}$  o conjunto dos  $n$  grupos formados na disciplina de PDS, por meio de sorteio foi definido quais grupos deveriam interagir entre si.

Inicialmente, para cada software  $s_z \in S$  foi sorteado um grupo  $g_t \in G$ , que devia representar em um primeiro momento o papel de adquirente do software  $s_z$ , sendo que em uma fase posterior um  $g_y \in G$  foi sorteado para atuar como fornecedor do software  $s_z$ , considerando as restrições impostas pelo grupo  $g_t$ .

A interação entre os grupos é dividida em quatro fases: (i) solicitar proposta; (ii) desenvolver proposta; (iii) avaliar proposta; e (iv) seminário final. A Figura 1 ilustra a sequência de atividades utilizada na APS, que serão descritas com mais detalhes a seguir, respectivamente nas subseções 3.1 a 3.4.

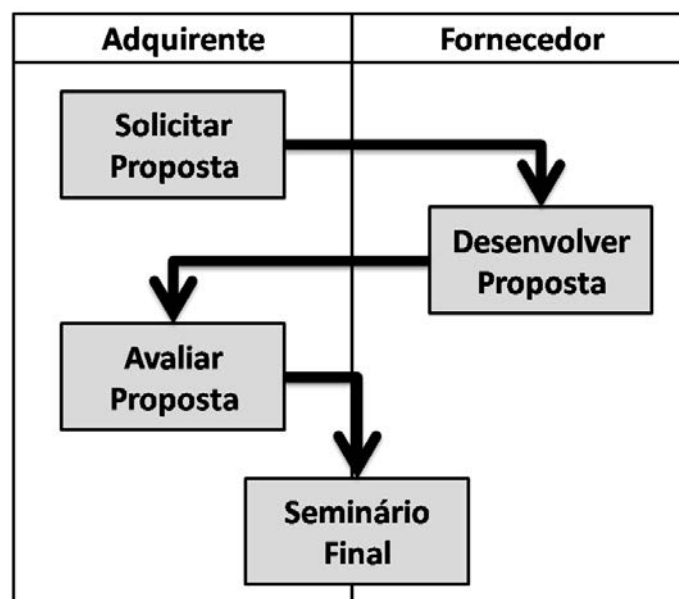


Figura 1. Fases da APS

### 3.1. Solicitar Proposta

Nessa fase, para cada um dos temas de software contidos no conjunto  $S$ , foi sorteado um grupo  $g_t \in G$ , que foi o responsável por redigir um pedido de proposta para software referente ao tema sorteado, atuando como adquirente.

O pedido de proposta desenvolvido devia contar obrigatoriamente com as quatro seguintes seções: (i) requisitos funcionais do software; (ii) restrições técnicas do software; (iii) responsabilidades dos envolvidos; e (iv) solicitação do protótipo de duas das telas principais do software.

Além de um nome, cada tema contido no conjunto  $S$ , contou com uma breve descrição, que devia ser usada como base para os grupos na redação de seus respectivos pedidos de proposta. Por exemplo, o tema software para gerenciamento de eventos, contou com a seguinte descrição: "Uma universidade precisa de um software para gerenciar os eventos que ela organiza. O software deve permitir o cadastro de cursos e palestras, e também gerar a programação com os horários do evento. Tal software também deve permitir a inscrição de alunos no evento, o controle de presença, e a emissão de certificados:

### 3.2. Desenvolver Proposta

Em uma data pré-definida, todos os grupos deviam entregar os seus respectivos pedidos de proposta, e então, para cada pedido de proposta foi definido por meio de sorteio outro grupo responsável por preparar uma proposta para o desenvolvimento do software solicitado.

Além dos requisitos funcionais e não funcionais do software, nessa fase, cada um dos grupos recebeu um conjunto de requisitos de qualidade de processo, que constam no modelo MPS.BR e também um modelo de processo tradicional (espiral ou prototipagem) que foram utilizados como base para a proposta de um modelo de processo.

Com relação aos requisitos de qualidade de processo, considerou-se apenas os níveis de maturidade G e F do MPS.BR, sendo que cada um dos grupos devia propor um processo de desenvolvimento baseado em um dos processos desses dois níveis (definido por sorteio). O nível G é composto pelos processos de Gerencia de Requisitos e Gerencia de Projetos, e o nível F contempla os processos de Aquisição, Gerencia de Configuração, Gerencia de Portfólio de Projetos, Garantia de Qualidade e Medição.

Destaca-se que tanto o modelo de processo que devia ser seguido, quanto o processo contemplado em dos dois níveis do MPS.BR, foram definidos por meio de sorteio para cada um dos grupos.

A proposta entregue nessa fase deveria contemplar cinco itens: (i) uma proposta de um processo de desenvolvimento de software, modelado com a notação BPMN (Business Process Model and Notation) [White 2008]; (ii) uma descrição textual de cada uma das atividades do processo; (iii) Uma matriz de rastreabilidade que associasse cada resultado esperado de um dos processos do guia MPS.BR com um uma ou mais atividades do processo proposto; (iv) cronograma e orçamento para o desenvolvimento do software; e (v) o protótipo de duas telas (conforme solicitado no pedido de proposta).



### 3.3. Avaliar Proposta

Cada grupo  $g_t$  encaminhou inicialmente um pedido de proposta a um grupo  $g_y$  (definido por sorteio), então na fase de avaliação o grupo  $g_t$  recebeu a proposta desenvolvida pelo grupo  $g_y$  e avaliou a conformidade dessa proposta com o respectivo pedido de proposta e também com os requisitos de processo (referente ao MPS.BR e a um dos modelos de processo tradicionais) que o grupo  $g_y$  adicionou a proposta.

Na avaliação cada grupo respondeu as quatro seguintes questões: (i) o modelo de processo sorteado foi seguido corretamente?; (ii) de acordo com o MPS.BR, todos os resultados esperados do processo sorteado para o grupo estão contidos no ciclo de vida proposto?; (iii) o cronograma e orçamento estão corretos?; e (iv) os protótipos propostos estão de acordo com o solicitado no pedido de proposta?

Cada uma das perguntas citadas devia ser respondida com sim ou não, sendo que, se a resposta fosse não, ela deveria ser justificada.

Para avaliar o desempenho dos grupos nessa fase, foi analisada a conformidade da avaliação dos grupos com a avaliação realizada pelo professor da disciplina, que respondeu as mesmas quatro questões para cada uma das atividades.

### 3.4. Seminário Final

No seminário final, cada um dos grupos apresentou o trabalho que desenvolveu em cada uma das fases, compartilhando assim com os outros grupos, o seu respectivo processo, os erros que foram encontrados durante a avaliação, dificuldades, a metodologia utilizada, entre outras informações.

Destaca-se que no seminário final, o trabalho foi dividido em três partes (de acordo com as fases anteriores da APS): (i) preparação do pedido de proposta; (ii) preparação da proposta; e (iii) avaliação da proposta.

No momento da apresentação, por meio de um sorteio, foi definido quais membros de cada um dos grupos, deveriam apresentar cada uma das partes da APS no seminário.

## 4. Avaliação dos Procedimentos Metodológicos

Dos 29 alunos que cursaram a disciplina em 2015, 27 deles aceitaram responder um questionário de avaliação da APS. Tal questionário contou com 4 perguntas, sendo as três primeiras objetivas, e a última dissertativa. Para cada uma das três primeiras questões os alunos deviam escolher entre cinco respostas (de acordo com a escala de Likert [Albaum 1997]): (i) discordo totalmente; (ii) discordo; (iii) nem concordo, nem discordo; (iv) concordo; e (v) concordo totalmente.

As questões contempladas no questionário foram as seguintes:

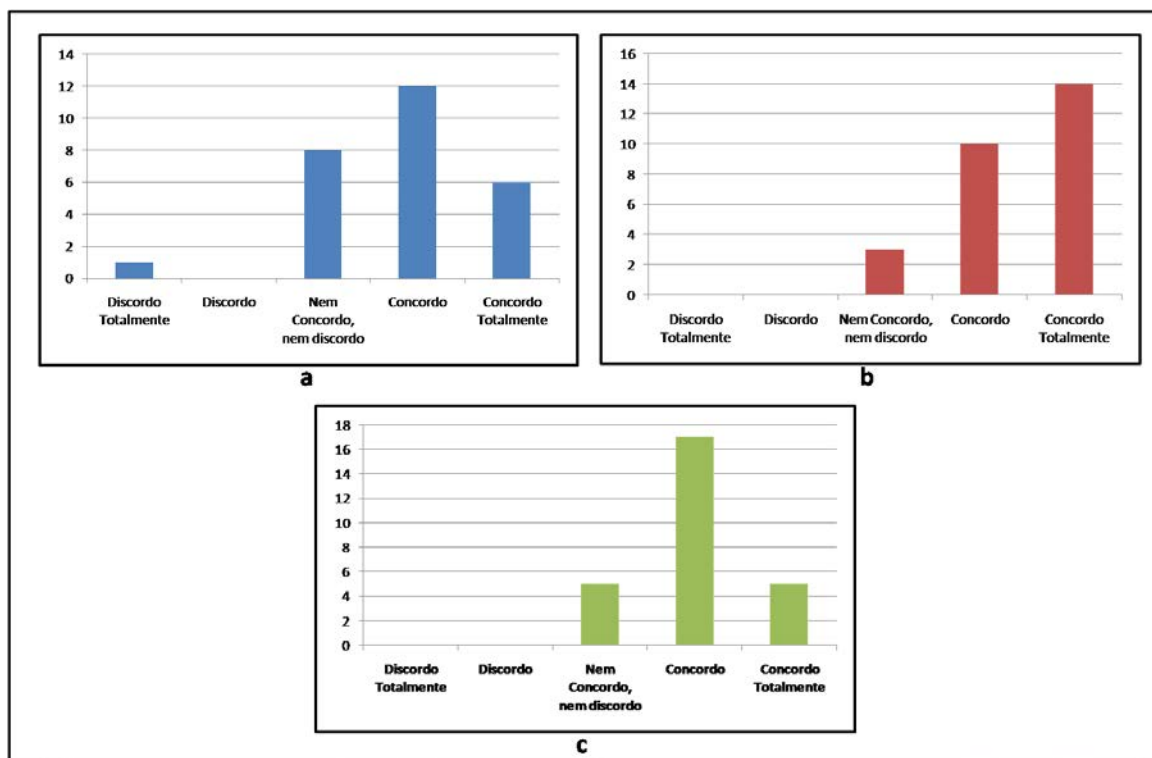
1. O fato de a APS proporcionar interações entre grupos diferentes foi um fator motivador para o desenvolvimento da APS?
2. A APS contribui para aquisição de novos conhecimentos em relação à disciplina?
3. A APS contribui para fixação dos conhecimentos apresentados durante as aulas?
4. Você tem alguma crítica ou sugestão que gostaria de compartilhar em relação a APS?

**Tabela 1. Resposta dos discentes ao questionário**

	Questão 1	Questão 2	Questão 3
Discordo Totalmente	1	0	0
Discordo	0	0	0
Nem concordo, nem discordo	9	3	5
Concordo	12	10	17
Concordo Totalmente	6	14	5

A Figura 2 apresenta em forma gráfica as respostas dadas as questões objetivas do questionário de avaliação. As mesmas informações são representadas na Tabela 1, na qual cada coluna é referente a uma das três primeiras questões, e as linhas representam o número de respostas em cada uma das alternativas.

Quanto a quarta questão, os alunos relataram que tiveram uma dificuldade maior na atividade de preparação do pedido de proposta em relação as demais atividades da APS (o que já era esperado), e também sugeriram uma etapa formal de interação direta entre os grupos.



**Figura 2. Respostas ao questionário: a) resposta a questão 1; b) resposta a questão 2; e c) resposta a questão 3.**

### 5. Análise dos resultados

Observa-se que de forma geral a opinião dos alunos em relação ao procedimento metodológico da APS foi positiva, pois em todas as perguntas objetivas a maioria dos alunos



disse concordar ou concordar totalmente com as afirmações contidas nas questões objetivas.

Com relação as perguntas relacionadas a contribuição da APS para aquisição e fixação de novos conhecimentos (respectivamente as questões 2 e 3), o número de alunos que concordaram ou concordaram totalmente foi maior que 80%, sendo que com relação à aquisição de novos conhecimentos, aproximadamente 89% dos alunos tiveram uma opinião favorável.

Destaca-se que apesar de alguns dos alunos terem respondido que nem concordavam nem discordavam, com exceção a resposta de um aluno na questão 1, nenhum dos discentes discordaram ou discordaram totalmente, das perguntas contidas no questionário.

Com relação às críticas e sugestões dos alunos as seguintes medidas já estão sendo tomadas na aplicação atual da APS (primeiro semestre de 2016):

- Quanto às dificuldades encontradas na redação da proposta, atualmente na nova aplicação dessa APS foi desenvolvido um vídeo que explica aos alunos o que deve ser feito em cada atividade, e também como eles serão avaliados (no vídeo é utilizado como exemplo uma das APS's entregues no semestre anterior). Espera-se que tal vídeo diminua as dúvidas dos alunos.
- Quanto à interação entre grupos, esperava-se que todos os grupos interagissem entre si, porém se observou que alguns deles não se comunicaram. Sendo assim, na aplicação atual da APS, será dedicado formalmente um período de meia aula, ao início da atividade de preparação da proposta, para que os grupos que representam o adquirente e o fornecedor de cada um softwares possam conversar sobre os requisitos, por meio de uma entrevista, tal como, comumente ocorre em processos reais de desenvolvimento de software.

## 6. Conclusão

Conclui-se que foram obtidos bons resultados com a aplicação da APS apresentada nesse trabalho. A maioria dos alunos se sentiu motivada no desenvolvimento do trabalho, o que como é apresentado por [Paiva 2011], é um grande desafio em disciplinas maciçamente teóricas, como é o caso da disciplina de PDS analisada nesse trabalho, que é ofertada no segundo semestre de um curso de graduação em Engenharia de Software.

Além de motivadora, de acordo com a opinião dos alunos, a APS também serviu para fixação e aquisição de conhecimentos, a qual provavelmente contribuiu para que o índice de aprendizagem efetiva da disciplina aumentasse, o que também é um dos desafios apresentados em [Paiva 2011].

Por fim, relata-se que devido aos bons resultados obtidos com a aplicação da APS, pretende-se continuar a aplicá-la em próximas turmas da disciplina de PDS.

## Referências

- Aguiar, Y. P. C., Saraiva, J., Dias, J., and Malheiros, Y. (2015). O ensino integrado de gestão de qualidade de software e engenharia de software aplicada como modelo de educação interdisciplinar: Um relato de experiência no ensino de graduação. In *Simpósio Brasileiro de Qualidade de Software*, pages 252–259.

- Albaum, G. (1997). The likert scale revisited: an alternate version. *Journal of the Market Research Society*, 39(2):331–348.
- Bavota, G., De Lucia, A., Fasano, F., Oliveto, R., and Zottoli, C. (2012). Teaching software engineering and software project management: an integrated and practical approach. In *Proceedings of the 34th International Conference on Software Engineering*, pages 1155–1164. IEEE Press.
- Cavalcante, N., Amâncio, F. D. d. S., Jucá, E. N., and Rodrigues, M. V. (2015). Uso de gamificação como auxílio para melhoria de processos: relato de experiência. In *Simpósio Brasileiro de Qualidade de Software*, pages 268–275.
- Gestal, P. R. E. and Barros, R. M. d. (2014). Proposta de um simulador para auxiliar no processo de ensino do scrum. In *Simpósio Brasileiro de Sistemas de Informação*, pages 723–736.
- ISO (2008). Iso/iec 12207 – tecnologia de informação-processos de ciclo de vida de software. *Rio de Janeiro: ABNT*.
- Kohwalter, T. C., Clua, E. W., and Murta, L. G. (2014). Reinforcing software engineering learning through provenance. In *Software Engineering (SBES), 2014 Brazilian Symposium on*, pages 131–140. IEEE.
- Kuhrmann, M., Fernández, D. M., and Münch, J. (2013). Teaching software process modeling. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 1138–1147. IEEE Press.
- Monsalve, E. S., do Prado Leite, J. C. S., and Werneck, V. M. B. (2015). Transparently teaching in the context of game-based learning: the case of simules-w. In *Proceedings of the 37th International Conference on Software Engineering-Volume 2*, pages 343–352. IEEE Press.
- Paiva, S. R. (2011). Uma revisão sistemática das pesquisas realizadas sobre a melhoria no ensino de engenharia de software. *Relatório Técnico - UFPB/CCEN*.
- Prikladnicki, R., Albuquerque, A. B., von Wangenheim, C. G., and Cabral, R. (2009). Ensino de engenharia de software: desafios, estratégias de ensino e lições aprendidas. *FEES-Fórum de Educação em Engenharia de Software*.
- Rodriguez, G., Soria, Á., and Campo, M. (2015). Virtual scrum: A teaching aid to introduce undergraduate software engineering students to scrum. *Computer Applications in Engineering Education*, 23(1):147–156.
- SOFTEX (2012). Mps.br - melhoria de processo do software brasileiro: Guia geral de software.
- Werneck, M., Nelson, M. A. V., and Alonso, E. (2012). Experiências de um trabalho interdisciplinar orientado por um processo de gerência de projetos em um curso de sistemas de informação. *FEES-Fórum de Educação em Engenharia de Software*.
- White, S. A. (2008). *BPMN modeling and reference guide: understanding and using BPMN*. Future Strategies Inc.