



INTER
FACES
CIENTÍFICAS

EXATAS E TECNOLÓGICAS

ISSN IMPRESSO - 2359-4934

ISSN ELETRÔNICO - 2359-4942

<http://dx.doi.org/10.17564/2359-4942.2018v3n2>

PRIORIZAÇÃO DE REQUISITOS ÁGEIS COM PDM

PRIORITIZATION OF AGILE REQUIREMENTS WITH PDM

PRIORIZACIÓN DE REQUERIMIENTOS ÁGILES CON PDM

Luís Antônio dos Santos Silva¹

Bruno Alves Reis Nascimento²

RESUMO

As empresas precisam desenvolver sistemas complexos com rapidez e eficiência. Priorização é uma parte crítica do desenvolvimento e isso envolve analisando a importância dos requisitos das partes interessadas e equipe de desenvolvimento e a seleção dos requisitos a serem implementadas em uma determinada versão do sistema. A decisão errônea sobre quais requisitos priorizar pode afetar o projeto e, conseqüentemente, sua qualidade perante os clientes. Nesta fase podem ocorrer problemas, porque o cliente pode acreditar que um determinado requisito

é mais importante do que ele realmente tem outro ou como você pode não dar importância a outros requisitos necessários. O principal objetivo desta pesquisa é propor um processo de priorização de requisitos para projetos de software ágeis baseados na técnica PDM, que foi originalmente proposta na área de planejamento estratégico

PALAVRAS-CHAVE

PDM, Priorização, Requisitos Ágeis

ABSTRACT

Companies need to develop complex systems quickly and efficiently. Prioritization is a critical part of development and this involves analyzing the importance of requirements by stakeholders and the development team, and selecting the requirements that will be implemented in a particular version of the system. The wrong decision about which requirements to prioritize can affect the project, and consequently its quality towards the clients. At this stage problems can occur because the customer may believe that a particular requirement has more importance than it actually has

of another or how it may not give necessary importance to another requirement. The main objective of this research is to propose a requirements prioritization process for agile software projects based on the PDM technique, which was originally proposed in the strategic planning area.

KEYWORDS

Agile Requirements. PDM. Prioritization

RESUMEN

Las empresas necesitan desarrollar sistemas complejos de forma rápida y eficiente. La priorización es una parte crítica del desarrollo y esto implica el análisis de la importancia de los requisitos por parte de los stakeholders y del equipo de desarrollo y la selección de los requisitos que se implementarán en una determinada versión del sistema. La decisión errónea sobre qué requisitos priorizar puede afectar el proyecto, y consecuentemente su calidad ante los clientes. En esta fase los problemas pueden ocurrir, ya que el cliente puede creer que un determinado requisito tiene más importancia de lo que realmente posee de otro

o cómo puede no dar importancia necesaria a otro requisito. El objetivo central de esta investigación es proponer un proceso de priorización de requisitos para proyectos de software ágiles basado en la técnica PDM, que fue originalmente propuesta en el área de planificación estratégica.

PALABRAS CLAVE

PDM, Software, Priorización de Requerimientos, Desarrollo ágil.

1 INTRODUÇÃO

Atualmente, softwares tem sido cada vez mais importante como instrumentos de apoio e tomadas de decisão nas mais diversas atividades. Dessa forma, as organizações agregam a tecnologia como uma fonte importante de inovação, fator crucial para o sucesso das empresas no tocante à competitividade e criação de valor (REIK; LINDERMANN, 2014). É possível observar esse aumento do software como apoio em várias áreas da atividade econômica (negócios, área financeira, construção civil, indústrias etc.) como em questões governamentais (saúde, educação etc.).

Melhorar os serviços prestados ou os produtos fabricados sempre é indispensável tanto para as empresas como para os governos, afinal empresas sempre precisam melhorar sua competitividade perante seus concorrentes e os governos melhorar seus serviços para atender de forma satisfatória sua população. Apesar da importância do software para as organizações, um dos grandes desafios refere-se ao fato de que, às vezes, o investimento feito em sistemas informatizados não fornece o retorno esperado, ou seja, os sistemas não se adequam à realidade das empresas onde são implantados.

Para Karlsson e Ryan (1996), um dos maiores riscos enfrentados por organizações que desenvolvem software comercial está associado ao não atendimento das necessidades e expectativas dos usuários. Isso claro, causa danos a imagem, perda de lucros e de credibilidade para as empresas.

Nessa realidade, analisar os requisitos do projeto e priorizá-los, de forma a obter a correta importância de cada um deles se torna essencial para o desenvolvimento de softwares que atendam a necessidade dos clientes.

Caracteriza-se assim o problema de pesquisa tratado por este artigo: como é possível obter uma estimativa do projeto desde as etapas iniciais e priorizar requisitos de forma que cada um receba o valor que lhe é devido?

Além da introdução, o artigo apresenta breve referencial teórico relacionado priorização de requisitos; a estimativa de software e estimativa de

tamanho, e o que várias fontes pensam a respeito sobre o assunto; abordagem metodológica sobre a técnica **Planning Poker** (PP) e sobre a técnica **Petic Decision Making** (PDM); abordagem metodológica sobre a união das técnicas ainda há pouco citadas; e por último, a conclusão.

2 PRIORIZAÇÃO DE REQUISITOS

Os softwares têm sido cada vez mais reconhecidos como essenciais em diversas atividades e apoio a decisões. Por isso, torna-se importantíssimo desenvolver softwares com menor custo e tempo possíveis ao identificar e priorizar os requisitos. Para Larman (2004), requisitos são capacidades e condições às quais o sistema – e de forma mais ampla, o projeto – deve atender. Young (2004) afirma que requisitos são atributos necessários em um sistema para que ele tenha valor e utilidade para os clientes e usuários.

Não é viável, considerando que são as necessidades dos usuários que movem o investimento em um projeto de software, que o foco do projeto seja outro senão a solução para essas necessidades, mas essa não é a realidade de muitos projetos que fracassam por causa disso e os requisitos são itens que os usuários demandam porque possuem valor em seu negócio, já que as prioridades são como eles esperam obtê-las, avaliando os diversos níveis de requisitos diante dos recursos limitados, já que se os recursos fossem ilimitados, não haveria a necessidade de priorização (GILB; MAIER, 2005), no entanto os recursos necessários ao desenvolvimento são limitados e devido a restrições de tempo e orçamento, pode ser difícil implementar todos os requisitos identificados para um sistema.

Os requisitos geralmente são implementados em etapas e a priorização ajuda a definir quais devem ser implementados prioritariamente (ALLEN *et al.*, 2008). Karlsson e Ryan (1996) e Berander (2004) apresentam visões semelhantes ao afirmarem que seleção do conjunto de requisitos é determinante para a satisfação dos usuários. Por isso, de acordo

com Duan e outros autores (2009), os desenvolvedores precisam ser cuidadosos na seleção do subconjunto dos requisitos que serão desenvolvidos. É necessário distinguir os requisitos que terão maior impacto para a satisfação dos usuários.

3 ESTIMATIVA DE SOFTWARE

A estimativa de software é uma atividade contínua que pode ser aplicada em todas as etapas do ciclo de vida do projeto. Com a estimativa, é possível coletar métricas que permitem prever a quantidade de pessoas, tempo e custos necessários para o desenvolvimento de um projeto de software, destaca Hazan (2008).

Pressman (2011) ressalta que se você não sabe para onde quer ir, qualquer caminho você pode seguir. Se você não sabe onde está, um mapa não irá lhe ajudar.

Quanto maior o prazo mais a incerteza e o conhecimento aumenta, quando isto ocorre, Bassi (2010) descreve que as estimativas devem ser refeitas, considerando um nível maior de detalhamento e assim, associando em probabilidades maiores de acerto.

As estimativas de custo e esforço em projetos de software são normalmente baseadas em prever o tamanho do sistema que será construído e é uma das mais críticas dentro do ciclo de vida de um projeto, pois será a partir dela que o projeto será posto em contrato e outras estimativas serão derivadas.

4 ESTIMATIVA DE TAMANHO

A estimativa de tamanho de software, segundo o Hazan (2008), é um processo pelo qual uma pessoa ou um grupo de pessoas estima o tamanho de um software. As estimativas de tamanho constituem a base para a derivação das estimativas de esforço, prazo e custos, descreve o CMMI (2012). Normalmente, o tamanho tem impacto na solução técnica e na gestão do projeto já que, estimativas imprecisas podem levar ao fracasso ou custos desnecessários de qualquer projeto.

Estimar o tamanho sobre um objeto, permitindo a interação entre os membros de uma equipe é uma boa

prática que promove maior assertividade no tamanho de um software, pois cada membro poderá possuir uma visão diferenciada deste objeto, destaca Cohn (2013). Técnicas como Triangulação, Analogia, **Planning Poker** e Desagregação possuem estas características, mas segundo Grenning (2002) e Cohn (2013) a mais conhecida é a técnica **Planning Poker**.

Diante do exposto, detalharemos a seguir a técnica **Planning Poker**.

5 PLANNING POKER

A técnica **Planning Poker** foi definida pela primeira vez por James Grenning em 2002 e popularizada por Mike Cohn no livro **Agile Estimating and Planning** no qual registrou o termo **Planning Poker** (PP). Cohn (2013) e Grenning (2002) descrevem o PP como uma técnica de estimativa de tamanho voltada para as metodologias ágeis de desenvolvimento de software.

O **Planning Poker** consiste-se em obter a estimativa por meio de um jogo de cartas, que, segundo Alves (2012), deve permitir que todos os membros da equipe de desenvolvimento (programadores, testadores, design e analistas) participem, colocando a sua visão de complexidade, levando em consideração o fator tempo e esforço para pontuar um cartão e após juntos chegar a um denominador comum na equipe por meio de consenso.

No **Planning Poker** cada integrante tem a sua disposição um baralho de 13 cartas, numeradas em uma sequência similar a encontrada nos números de Fibonacci. As cartas contém os tamanhos de 0, ½, 1, 2, 3, 5, 8, 13, 20, 40 e 100 que serão atribuídos a um cartão, havendo ainda uma carta com o símbolo de interrogação no qual representa que o estimador não está apto a estimar e outra carta com a imagem de uma xícara de café no qual representa a sugestão de uma pausa.

Durante o **Planning Poker** devem ser realizadas rodadas para obter a estimativa de um cartão que possui uma estória ou tarefa a desenvolver. As diferenças que surgirem durante as rodadas deverão ser mediadas por um coordenador, que no Scrum este papel é de responsabilidade do Scrum Master.

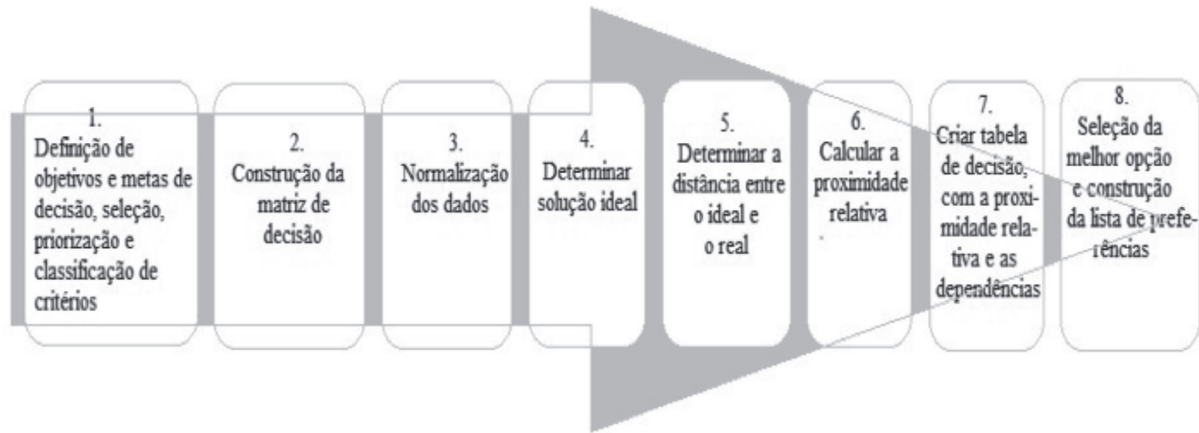
O **Product Owner**, é o responsável por explicar o que deverá ser desenvolvido, sendo um importante papel para retirar possíveis dúvidas a respeito do cartão, evitando assim, o retrabalho.

6 SOLUÇÃO PETIC DECISION MAKING (PDM)

Rocha (2017) afirma que o **Petic Decision Making** (PDM) é um modelo de apoio à tomada de decisão multicritério que objetiva a redução de incertezas no processo de ordenação e seleção de investimentos, de for-

ma a não possibilitar dupla interpretação na indicação de opção considerada preferível. O PDM tem como base o TOPSIS, o qual possui um fluxo de trabalho que parte da elaboração dos objetivos de decisão, concluindo na listagem ordenada de preferências. Nesse fluxo, exposto a seguir, as etapas 1, 7 e 8, além do ciclo contínuo, são as adaptações relativas ao PDM, criadas para atender as necessidades do planejamento estratégico, bem como para a análise de dependência, fator essencial para o sucesso da seleção de ações. Já as etapas de 2 a 6 são herdadas diretamente do TOPSIS.

Figura 1 – Ciclo de vida de atividades do PETIC Decision Making



Fonte: Rocha (2017)

O objetivo claro do processo de priorização é a redução de custo e a ampliação de benefícios. Assim, são empregados 10 critérios mais o critério de dependência que podem ser alterados, expandindo-se ou reduzindo-se de acordo com os objetivos de decisão, mas o critério de dependência é obrigatório para o PDM. Os critérios são expostos no Quadro 1.

Quadro 1

ID	Critério	Tipo	Atributos
C1	ROI	Benefício	0-4

ID	Critério	Tipo	Atributos
C2	Exigência de Norma / Lei	Benefício	0-4
C3	Evita Colapso na Infraestrutura	Benefício	0-4
C4	Alinhamento Estratégico	Benefício	0-4

ID	Critério	Tipo	Atributos
C5	Principais stakeholders beneficiados	Benefício	0-4
C6	Custo	Custo	Valor Monetário
C7	Riscos Envolvidos	Custo	0-4
C8	Probabilidade de Entrega	Benefício	0-4
C9	Esforço	Custo	0-4
C10	Otimização	Benefício	0-4
Cdep	Dependência	Dependência	Quantidade de dependências

Fonte: Rocha (2017).

7 UNIÃO ENTRE PLANNING POKER E PDM

Anteriormente foi explicado o que é o **Planning Poker (PP)** e **Petic Decision Making (PDM)**, e como cada um deles funciona. Agora, iremos unir essas duas técnicas para a priorização de requisitos dos projetos.

Primeiro mudaremos os critérios para que o PDM se adeque a priorização de requisitos. Nessa abordagem os critérios serão Importância, Facilidade, Custo e Dependências. Adiante os critérios são expostos no Quadro 2.

Quadro 2

ID	Critério	Tipo	Atributos
C1	Importância	Benefício	Valores das cartas do Planning Poker

ID	Critério	Tipo	Atributos
C2	Facilidade	Benefício	Valores das cartas do Planning Poker
C3	Risco	Custo'	Valores das cartas do Planning Poker
Cdep	Dependência	Dependência	Quantidade de dependências

Fonte: Elaborado pelos autores.

Assim, propomos que na primeira etapa do PDM, o **Product Owner** pontue primeiro a importância do requisito. Na segunda etapa, o **Planning Poker** entra em ação, para que no final do seu processo possa pontuar o restante dos critérios (facilidade e risco) e na terceira etapa o time de desenvolvimento define as dependências de cada um dos requisitos. Vale ressaltar, que quanto maior for o valor dado à importância, mais importante o requisito se torna; quanto maior for o valor dado a facilidade, mais fácil será o requisito a ser desenvolvido; quanto maior for o valor dado ao risco, maior será o cuidado que o time de desenvolvedores deve ter para implementar o requisito; quanto maior for o valor dado a dependência, mais ele precisará de outros requisitos para que funcione de forma adequada.

Primeiramente, quatro etapas são incorporadas ao PDM e é retirada a primeira etapa (Definição de objetivos e metas de decisão, seleção, priorização e classificação de critérios) e a última (Seleção da melhor opção e construção da lista de preferências) mostrada na Figura 1 para utilizá-lo com a finalidade de priorização de requisitos.

O fluxo do PDM segue normalmente até a última etapa onde nela é criada o **backlog** dos requisitos já priorizados. Logo adiante é mostrada a Figura 2 que apresenta o ciclo de vida da união das duas técnicas.

Figura 2 – Ciclo de vida de atividades da união do Planning Poker com o PETIC Decision Making



Fonte: Elaborado pelos autores.

8 CONCLUSÃO

Um dos maiores problemas a respeito dos projetos de desenvolvimento de software refere-se a sua não conformidade aos requisitos desejados pelos usuários. Considerando essa dificuldade, este artigo apresentou uma abordagem da união de duas técnicas para melhorar a priorização de requisitos por parte dos usuários e do time de desenvolvimento.

Acredita-se que a satisfação dos usuários seja preponderante para o sucesso de um projeto de e a continuidade de utilização de um software. Por isso, a abordagem buscou o alinhamento entre as necessidades dos usuários e o software criado.

Acreditamos que essa abordagem permite aumentar as chances de sucesso dos projetos de desenvolvimento e elevar a qualidade dos produtos de software.

REFERÊNCIAS

ALVES, S.W.D. **Ferramenta para apoio à estimativa baseada em Planning Poker utilizando a**

metodologia scrum. Recife: UFPE. Disponível em: <<http://www.cin.ufpe.br/~tg/2012-1/dwas.pdf>>. Acesso em: 14 abr. 2018.

BASSI, F.D.L. **Experiências com desenvolvimento ágil.** São Paulo. Disponível em: <http://www.teses.usp.br/Dissertacao_Metodos_Ageis_Dairton_Bassi.pdf>. Acesso em: 14 abr. 2018.

CMMI – Capability Maturity Model Integration. CMMI for development: CMMI-DEV, Pittsburgh. **Software Engineering Institute**, v.1. n.3, 573p. 2012.

COHN, M. **Agile Estimating and Planning.** 2013. Disponível em: <<http://www.mountangoatsoftware.com/books/agile-estimating-and-planning>> e <<http://www.mountangoatsoftware.com/blog/estimating-with-tee-shirt-sizes>>. Acesso em: 16 abr. 2018.

GRENNING, J. **Planning poker or how to avoid analysis paralysis while release planning.** 2002. Disponível em: <<http://renaissancesoftware.net/files/articles/PlanningPoker-v1.1.1.pdf>>. Acesso em: 7 abr. 2018.

HAZAN, C. Análise de pontos de função: uma aplicação nas estimativas de tamanho de projetos de Ssoftware. **Engenharia de Software**, Rio de Janeiro, p.25-30, 2008.

KARLSSON, J.; RYAN, K. Supporting the selection of software requirements. **International Workshop on Software Specification and Design (IWSSD '96)**, 8th. Proceedings. 1996, p.146-149.

LARMAN, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos e ao Processo Unificado.** 2.ed. Porto Alegre: Bookman, 2004. 607p.

PRESSMAN, Roger S. *et al* (Rev.). **Engenharia de software: uma abordagem profissional.** 7.ed. São Paulo: McGraw Hill, 2011.

REIK, Alexander; LINDERMANN, Udo. Analyzing which technology-related information is required at which stage of product planning: a literature-based approach to integrate technology management into strategic product planning. **Engineering, technology and innovation**, x., 2014, Cidade. Proceedings Bergamo: IEEE, 2014. p.1-10.

ROCHA, Fábio Gomes. **Petic Decision Making (PDM)**: um modelo automatizado para apoio à tomada de decisão estratégica em TIC. Sergipe: UFS. Disponível em: <<https://ri.ufs.br/handle/riufs/3386>>. Acesso em: 12 abr. 2018.

YOUNG, R.R. The requirements engineering handbook. Boston: Artech House, 2004. 251p.

Recebido em: 3 de Março de 2018
Avaliado em: 24 de Abril de 2018
Aceito em: 7 de Maio de 2018

1 Bacharel em Sistemas da Informação Unit, membro do GPITIC.
Email: luisopentec@gmail.com.

2 Técnico em Informática – Senai, tecnólogo Gestão de Tecnologia da Informação – Estácio, Mestrando em Ciências da Computação – UFBA.
Email: b.reisnascimento@gmail.com.