



FLEXIBILIDADE NO DESENVOLVIMENTO DE SOFTWARE: A PERCEPÇÃO DE GERENTES DE PROJETO DE ARACAJU/SE

FLEXIBILITY IN SOFTWARE DEVELOPMENT: THE PERCEPTION OF PROJECT MANAGERS ARACAJU / SE

Michell Angelo Santos Lima* E-mail: michell.aju@gmail.com

Maria Conceição Melo Silva Luft* E-mail: ceica@ufs.br

*Universidade Federal de Sergipe (UFS), Aracaju, SE

Resumo: Os gerentes de projetos de desenvolvimento de *software* precisam cada vez mais lidar com fatores como a competição, demanda por qualidade, menores ciclos de vida de produtos e mudanças tecnológicas constantes no mercado. Com o propósito de contribuir para tal discussão, esta pesquisa investiga como as características de flexibilidade são percebidas por gerentes de projetos no desenvolvimento de *software* através da caracterização dos modelos de processos de desenvolvimento de *software* adotados no contexto estudado, assim como no destaque de elementos de flexibilidade nestes processos. A metodologia teve caráter exploratório, descritivo e qualitativo e adotou-se a estratégia de pesquisa estudo de casos múltiplos, na qual os casos foram caracterizados pelos processos de desenvolvimento de *software* e, como fonte de evidências, escolheram-se os gestores de projetos de *software* que atuam em empresas de Aracaju há mais de cinco anos na função, por considerar que estes profissionais possuem uma visão mais abrangente do processo de desenvolvimento de tais sistemas. Como resultados obtidos, os gerentes de projetos, apesar de focarem nos aspectos técnicos do processo de desenvolvimento de *software*, associam o nível de sucesso do projeto a outros fatores, tais como a flexibilidade com relação ao cliente, nos grupos de trabalho e no processo de desenvolvimento.

Palavras-chave: Desenvolvimento de *Software*. Flexibilidade. Processos. Grupos de Trabalho. Participação do Cliente.

Abstract: Managers of software development projects increasingly need to deal with factors such as competition, demand for quality, lower product life cycles and constant technological changes in the market. Aiming to contribute to this discussion, this research investigates how the characteristics of flexibility are perceived by project managers in software development through the description of process models for software development adopted in the context studied, as well as highlighting elements flexibility in these processes. The methodology was exploratory, descriptive as well as qualitative, and adopted the strategy of researching a multiple case study in which the cases were characterized by the processes of software development. As a source of evidence, were chosen managers of software projects on companies that operate in Aracaju more than five years in the role, considering that these professionals possess a thorough overview of the development process of such systems. As results, project managers, while focusing on the technical aspects of the software development process, associate the level of success of the project to factors, such as flexibility with respect to customer in working groups and in the process development.

Keywords: Software Development. Flexibility. Processes. Work Groups. Customer Participation.

1 INTRODUÇÃO

A engenharia de *software* é baseada em um conjunto de princípios, tecnologias e processos para o desenvolvimento de *software* que busca aumentar a capacidade de produção, melhorar processos e produzir sistemas mais complexos (BASILI, 1992; SOMMERVILLE, 2007). Para tanto, são utilizadas metodologias de desenvolvimento, que consistem em um conjunto de atividades e resultados associados que embasam a produção de *software* (SOARES, 2003).

Diante da complexidade inerente aos grandes sistemas de *software*, inclusive os sistemas empresariais, exigem-se novas técnicas e métodos para controlar as implicações decorrentes do grande fluxo de dados e informações. Dessa forma, existem vários modelos com técnicas e formas distintas que determinam um conjunto de atividades, ações, tarefas e ferramentas de trabalho, formando um protocolo necessário para a obtenção de resultados de alta qualidade (PRESSMAN, 2006; SOMMERVILLE, 2007).

Os primeiros modelos de processos de engenharia de *software*, denominados clássicos ou tradicionais (BOEHM, 2002), são caracterizados pelo uso extenso de planejamento e processos codificados almejando uma atividade eficiente e previsível ao antecipar por completo os requisitos de um sistema (HIGHSMITH; COCKBURN, 2001; BOEHM, 2002), e prezar por processos de engenharia padronizados, controláveis e previsíveis (MOE; DINGSØYR; DYBÅ, 2010).

No entanto, vale ressaltar que esses modelos de processos foram desenvolvidos há algumas décadas, em um ambiente de negócios diferente do atual, no qual até o acesso aos computadores era limitado (SOARES, 2003). Com a dinâmica do mercado atual, com características tais como competitividade, demanda por qualidade, menores ciclos de vida de produtos e mudanças tecnológicas constantes (CHESBROUGH, 2012), surgiram inúmeros modelos de processos diferentes decorrentes da insatisfação dos pesquisadores e profissionais, propiciando a adequação dos modelos de processos de acordo com as exigências presentes no ambiente de negócios (ABRAHAMSSON *et al.*, 2003).

Dentro deste contexto, surgiram os processos ágeis, que buscam sanar a necessidade de construir e entregar sistemas complexos de forma rápida, diminuindo problemas existentes em relação a requisitos imprevisíveis, ambiente de

negócios dinâmico e o surgimento de novas tecnologias (LAN; RAMESH, 2008), além de conservar a qualidade do produto, uma vez que o mercado exige inovação e alta qualidade (HIGHSMITH; COCKBURN, 2001).

Em resumo, os fundamentos dos processos ágeis caracterizam-se pela adoção da mudança visando atender os valores do cliente, o planejamento e entrega de *software* freqüente, a centralização dos recursos humanos envolvidos no processo de desenvolvimento, a excelência técnica e a colaboração do cliente em todas as etapas de elaboração (WILLIAMS; COCKBURN, 2003; SIDKY; ARTHUR, 2007).

Nessa perspectiva dos processos ágeis, é necessária a percepção por parte dos gestores de projetos de que o *software* ou seu desenvolvimento faz parte de um sistema sociotécnico, ou seja, composto por entidades humanas e técnicas que funcionam como um conjunto integrado e coordenado (DYNGSOYR; DYBA, 2012), no qual fatores humanos (sociais e organizacionais) são críticos no cumprimento com sucesso dos objetivos do *software* (CHERRY; ROBILLARD, 2008).

Diante dos modelos de processos de desenvolvimento de *software* existente e dos aspectos humanos envolvidos nestes processos, é necessário compreender as dificuldades que ainda são encontradas no desenvolvimento de *software* no cenário das empresas que desenvolvem estes produtos. Para tanto, este artigo investiga como as características de flexibilidade são percebidas por gerentes de projetos no desenvolvimento de *software* através da caracterização dos modelos de processos de desenvolvimento de *software* adotados no contexto estudado, assim como no destaque de elementos de flexibilidade nestes processos. Sendo assim, tem como objeto de estudo os processos de desenvolvimento de *software* em empresas no município de Aracaju, SE, e como fonte de evidência os gestores de projetos destes sistemas.

2 REFERENCIAL TEÓRICO

2.1 Processos de Desenvolvimento de Software

O desenvolvimento de *software* é baseado em modelos de processos da engenharia de *software*, os quais fornecem formas distintas e têm suas próprias

técnicas (SOMMERVILLE, 2007). Estes processos definem um conjunto distinto de atividades, ações, tarefas, marcos e ferramentas de trabalho, formando um roteiro útil necessário para resultados de alta qualidade (PRESSMAN, 2006). Neste tópico discute-se os processos denominados tradicionais e ágeis.

2.1.1 Processos tradicionais

Os primeiros modelos de processos de engenharia de *software* foram os denominados clássicos ou tradicionais (BOEHM, 2002), que partem do princípio de que é possível eliminar mudanças nos processos de desenvolvimento quando há um esforço em antecipar por completo os requisitos de um sistema (HIGHSMITH; COCKBURN, 2001). Vale ressaltar que na época em que foram criados tais métodos, na década de 1970, o custo de fazer alterações e correções era muito alto, já que o acesso aos computadores era limitado e não existiam modernas ferramentas de apoio ao desenvolvimento do *software* (SOARES, 2003).

Assim, os processos tradicionais surgiram em um contexto de desenvolvimento de *software* muito diferente do atual, baseado apenas em um mainframe e “terminais burros”, ou seja, terminais que não interagem com outras máquinas (ROYCE, 1970). Nesse sentido, a perspectiva tradicional no desenvolvimento de *software* está enraizada no paradigma racionalista, que promove uma abordagem orientada ao planejamento de desenvolvimento de *software* usando um processo de engenharia padronizado, controlável e previsível (MOE; DINGSØYR; DYBÅ, 2010).

2.1.2 Processos ágeis

Em virtude das considerações feitas anteriormente, percebe-se que os processos tradicionais têm uso recomendado apenas em situações em que os requisitos do sistema são estáveis e requisitos futuros são previsíveis (SOARES, 2003). No entanto, diante do contexto atual de desenvolvimento e como resposta à ineficiência dos métodos de desenvolvimento de *software* existente no final da década de 1990, surgiram inúmeros processos seguindo a perspectiva ágil, que

possibilitaram a adequação desses processos para as diferentes situações (ABRAHAMSSON et al., 2003).

Os processos ágeis se referem a um grupo de métodos de desenvolvimento de *software* que se baseiam em ciclos iterativos e incrementais, cuja finalidade é construir e entregar produtos de *software* de forma rápida (SOMMERVILLE, 2007), amenizar problemas existentes no desenvolvimento de sistemas quanto aos requisitos imprevisíveis, ambiente de negócios dinâmico e o surgimento de novas tecnologias que tem impulsionado as demandas por sistemas complexos (LAN; RAMESH, 2008).

Tais mudanças ambientais externas causam variações críticas e inevitáveis no *software*. Desse modo, o desafio para os desenvolvedores é a capacidade de reagir a essas mudanças, reduzir o custo e diminuir o tempo de resposta (HIGHSMITH; COCKBURN, 2001; WILLIAMS; COCKBURN, 2003; ABBAS; GRAVELL; WILLS, 2008), considerando que os detalhes de requisitos são muitas vezes difíceis de serem explicados sem um primeiro contato com as funcionalidades do sistema em evolução (BECK, 2000). Assim, não se busca eliminar o retrabalho, nem apenas acomodar as mudanças, mas também manter a qualidade do *software*, já que o mercado exige inovação e *software* de alta qualidade (HIGHSMITH; COCKBURN, 2001).

2.2 Flexibilidade

A globalização trouxe um ambiente de rápidas mudanças tecnológicas, afetando diretamente os *softwares* utilizados diariamente pelas pessoas, mantendo cada elemento dos sistemas de informação em um estado de constante fluxo de mudanças: no *hardware*, nos sistemas operacionais e em *softwares* cooperativos (EDEN; MENS, 2006).

Desse modo, o modelo de produção na sociedade do conhecimento tem como característica principal a flexibilidade, que influencia o modo pelo qual os profissionais são formados para trabalhar nesta sociedade, a qual tem como características essenciais a flexibilidade e a criatividade (CAVALCANTI; GOMES, 2001).

Esse ambiente também envolve possibilidades de alteração durante o processo de desenvolvimento, tornando-o imprevisível e complexo por exigir uma flexibilidade capaz de responder rapidamente às mudanças (SCHWABER, 2004). Aliado a isso, as expectativas do cliente também motivam melhorias de desempenho, correções de *bugs*, falhas de segurança e as tentativas para montar uma cooperação cada vez mais sofisticada nos sistemas de *software* (EDEN; MENS, 2006).

Além disso, a flexibilidade é um aspecto que possui bastante destaque nos processos de desenvolvimento de *software* mais recentes, que passaram a enfatizar princípios como interação com o **cliente** em todas as fases do projeto, com alterações em todas estas fases e auto-organização da **equipe** de acordo com o **projeto** que está sendo desenvolvido (BECK *et al*, 2001), tal como destaca-se a seguir.

2.2.1 Flexibilidade com relação ao cliente

A flexibilidade de organizações de produtos e serviços como objeto de estudo vem crescendo nas últimas décadas na comunidade acadêmica (CHOU; TEO; ZHENG, 2008). Ainda assim, o conceito de flexibilidade é bastante discutido e mal definido uma vez que este termo está relacionado a múltiplas áreas de conhecimento e abrange várias dimensões dentro de cada área. Na literatura, é possível identificar vários tipos de flexibilidade e cada tipo apresenta nomenclaturas diferentes e definições divergentes (SETHI; SETHI, 1990; BENG, 2009).

No tocante à flexibilidade na interação com o cliente e suas demandas, Sethi e Sethi (1990) a denomina flexibilidade interna e a define como habilidade relativa à lida com o cliente e o atendimento de suas solicitações de forma eficiente.

2.2.2 Flexibilidade nos grupos de trabalho

Por flexibilidade do grupo de trabalho entende-se a habilidade do grupo em ajustar suas atividades de acordo com as condições de mudança sem que estes ajustes resultem em uma desorganização do grupo (SETHI; SETHI, 1990), de modo

que as funções desempenhadas por trabalhadores podem ser alteradas em resposta às variações do projeto (BENG, 2009).

Nesse aspecto, Trkman (2010) ressalta um importante aspecto da flexibilidade dentro das equipes de trabalho: a contratação de especialistas ou de generalistas. O autor explica que os especialistas constroem uma rotina mais rápida e possuem um conhecimento mais profundo do que os generalistas, resultando em um trabalho mais rápido e de melhor qualidade. No entanto, os generalistas trazem mais flexibilidade ao processo, uma vez que suas funções são remanejadas mais facilmente (TRKMAN, 2010), conforme a demanda solicitada em cada momento.

2.2.3 Flexibilidade nos processos no desenvolvimento de *software*

Com a pressão crescente de um mercado altamente dinâmico e competitivo no desenvolvimento rápido de *software*, os fatores que fazem os projetos de desenvolvimento de *software* bem sucedidos tem atraído muitos pesquisadores e profissionais (GÜNSEL *et al.*, 2012), haja vista que a capacidade de resposta às mudanças no mercado durante o desenvolvimento de *software* é essencial na obtenção de um produto de qualidade (LI *et al.*, 2010).

Assim sendo, existem estudos que apontam que a qualidade do produto de *software* sofre uma influência positiva da flexibilidade em seu desenvolvimento. Atualmente, existem várias diretrizes de desenvolvimento de *software* e metodologias diferentes para aumentar a flexibilidade do produto de *software*, cabendo à gestão escolher metodologias para promover essa qualidade (LIU *et al.*, 2008), de acordo com o projeto.

3 PROCEDIMENTOS METODOLÓGICOS

Este estudo pode ser classificado como exploratório e descritivo. É exploratório por considerar que os trabalhos que abordam aspectos não técnicos, como flexibilidade no desenvolvimento de *software*, ainda são incipientes e que as discussões acerca dessas atividades ainda têm muito que avançar para possibilitar uma maior compreensão do fenômeno estudado. É também descritivo ao considerar

que esse estudo pretende descrever como são percebidas as características de flexibilidade no desenvolvimento de *software*.

A pesquisa se enquadra ainda como qualitativa, uma vez que busca, por meio de relações subjetivas, estabelecer interpretações acerca do fenômeno estudado, como também pelo fato de a pesquisa qualitativa permitir várias possibilidades de estudar os fenômenos mais complexos por meio de aspectos humanos (CRESWELL, 2003).

Como estratégia de pesquisa foi adotada o estudo de caso, uma vez que o objetivo da pesquisa é realizar um estudo sobre as características de flexibilidade percebida pelos gerentes de projeto no desenvolvimento de *software*. Ademais, considerando que a flexibilidade no desenvolvimento de *software* é um aspecto ainda carente de investigação e que necessita ampliar a área de conhecimento no que concerne à capacidade de ajustes frente às condições de mudanças impostas; e por se tratar de um tema atual, no qual ainda existe discussão acerca de como esses processos funcionam frente às mudanças constantes no *software* em decorrência de um mercado globalizado, e das características de volatilidade do próprio *software*, escolheu-se o estudo de casos múltiplos.

O campo de desenvolvimento deste estudo foram cinco ambientes de desenvolvimento de *software* que adotam processos de desenvolvimento localizados no município de Aracaju, SE, conforme apresenta o Quadro 01.

Quadro 01 - Síntese dos casos de estudo

Casos de estudo	Segmento de negócio da organização	Tipo de modelo de processo adotado
Processo Alfa	Instituição Financeira Estatal	Tradicional
Processo Beta	Desenvolvimento de aplicativos para Internet sob Demanda	Ágil
Processo Delta	Instituição de Ensino Superior Privada	Híbrido
Processo Gama	Desenvolvimento de aplicativos para dispositivos móveis	Ágil
Processo Zeta	Instituição Financeira Estatal	Tradicional

Fonte: Elaborado pelo autor (2013)

Para explorar melhor o assunto pesquisado, optou-se pela entrevista semi-estruturada, a fim de buscar compreender, por meio das interações subjetivas demonstradas pelo gestor, a percepção do mesmo acerca da flexibilidade nos processos de desenvolvimento de *software*. As questões da entrevista foram

Revista Produção Online, Florianópolis, SC, v.14, n. 3, p. 1050- 1076, jul./set. 2014.

definidas a partir das categorias de análise: modelos de processos de desenvolvimento de *software*; flexibilidade com relação ao cliente; flexibilidade no grupo de trabalho; flexibilidade no processo de desenvolvimento de *software*.

Inicialmente foi realizada uma entrevista piloto com uma gerente de projetos de desenvolvimento de *software*, a fim de validar o roteiro de entrevista e verificar se as questões dispostas conseguiram capturar os elementos necessários à pesquisa, como também se estas foram dispostas de forma clara ao entrevistado, servindo de subsídio à adequação do roteiro.

As entrevistas foram realizadas no período de 03 de fevereiro à 11 de março de 2013, sendo estas gravadas e transcritas com o objetivo de proceder a sua análise. Para sua interpretação foi utilizada a técnica de análise de conteúdo, que consiste na utilização de procedimentos sistemáticos e objetivos de descrição do conteúdo das mensagens. Sendo assim, o interesse não está somente na descrição dos conteúdos, mas sim no que poderá ser aprendido após serem tratados (BARDIN, 2008).

4 APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

4.1 Modelos de processos no desenvolvimento de *software*

No tocante aos tipos de sistemas produzidos pelas empresas, observou-se no Quadro 02 que os entrevistados gerenciam equipes que produzem vários tipos de sistemas. Apesar de não ser um dos elementos de análise, percebeu-se a importância da atividade fim da organização para a interpretação dos achados desta pesquisa. Dentre os entrevistados, os gestores dos processos Alfa, Delta e Zeta desenvolvem estes produtos para satisfazerem as necessidades da própria organização, de forma que a produção de *software* não é a atividade fim da organização.

Ainda nestes casos, observou-se que os ciclos de vida dos projetos são mais longos (médio e grande porte) e adotam processos tradicionais (casos Alfa e Zeta) ou híbrido (caso Delta), que consiste em adaptações de modelos de processos de desenvolvimento de *software* que já existem no mercado.

O caso Processo Alfa foge dos padrões quanto aos modelos de processos, por apresentar o uso de modelo tradicional e ágil no mesmo ambiente caracterizando a coexistência de modelos de processos. Esta coexistência de modelos de processos é destacada por Vinekar, Slinkman e Nerur (2006), que indicam que algumas organizações têm mantido a utilização dos dois tipos de modelos de processos, com o objetivo de desfrutar das vantagens proporcionadas por cada modelo. Nesse sentido, no caso Alfa é adotada a metodologia ágil para pequenas melhorias e ajustes em produtos de *software* em andamento ou já concluídos especificamente, por ser projetos com menor ciclo.

Já nos outros dois casos restantes na pesquisa (Beta e Gama), ainda no Quadro 02, apontou-se que estes processos têm como característica em comum funcionarem em fábricas de *software*, possuem ciclos de vida curtos e os processos adotados serem baseados na metodologia ágil denominada *Scrum*.

Quadro 02 - Caracterização do desenvolvimento de *software*

Elementos/Casos	ALFA	BETA	DELTA	GAMA	ZETA
Tipos de software/sistemas desenvolvidos	Sistemas de crédito, administrativo e sistemas de serviços bancários.	Sistemas para internet voltados para educação à distância, gestão de empresas, portais de conteúdos e dispositivos para celulares.	Sistemas educacionais, financeiros, gestão, recursos humanos e intranet.	Aplicativos <i>Mobile</i>	Sistemas de Canais de Automação Bancária, Gestão de Contas, Aplicações Financeiras, Gestão de Cliente e Estratégicos de Business Intelligence (BI)
Classificação de escopo dos projetos e duração.	Médio e grande porte: mais de 30 dias; Pequeno porte: menos de 30 dias.	Pequeno porte: Com duração de seis meses e divididos em pequenos projetos com ciclos de duas semanas.	Médio e grande porte: 5 meses a um ano; Pequeno porte: duas semanas.	Pequeno: Duração de 15 a 90 dias.	Grande Porte: Duração entre 12 e 18 meses.
Modelos de processo utilizados	Projetos de médio e grande porte: Tradicional (RUP); Projetos de pequeno porte: Ágil (<i>Scrum</i>).	Método Ágil (<i>Scrum</i>)	Método Híbrido Não adota um modelo definido de mercado. Criou um modelo a partir dos modelos existentes adaptado às necessidades organizacionais	Método Ágil (<i>Scrum</i>)	Tradicional (RUP)

Fonte: Pesquisa de Campo (2013)

Ainda em relação ao tipo de processo adotado pelos entrevistados dos casos processos Beta e Gama, isso se dá porque a adoção destes modelos de processos ágeis é interessante para empresas que possuem projetos pequenos e necessitam competir no mercado através do lançamento de produtos inovadores, pois o mercado de *software* exige produtos inovadores e de alta qualidade (HIGHSMITH; COCKBURN, 2001; BOEHM, 2002; ABRAHAMSSON et al., 2003; LEE; XIA, 2010; CHESBROUGH, 2012).

No que concerne ao modelo específico de processo adotado pelos gerentes entrevistados, observa-se que quando o modelo tradicional (casos Alfa e Zeta) adotou-se o *Rational Unified Process* (RUP) e quando o modelo é ágil, adotou-se o *Scrum*.

4.2 Flexibilidades com relação ao cliente

Em relação ao nível de participação do cliente no processo, no Quadro 03, observou-se que todos os entrevistados ressaltam que o cliente participa das fases de solicitação, definição dos requisitos, homologação e validação do *software*.

Nos processos Beta e Gama, os gerentes apontaram que os clientes acompanham todo o processo. Vale destacar que estas empresas são fábricas de *software*, ou seja, sua atividade fim é produção destes sistemas, como também o fato delas adotarem processos ágeis, que preza pela participação dos clientes em todas as etapas do processo (BECK *et al.*, 2001).

Ainda no Quadro 03, é possível apontar que os gestores entrevistados destacaram a importância do cliente no processo de desenvolvimento de *software*. No entanto, quanto aos aspectos positivos e negativos da participação do cliente no processo de desenvolvimento, para alguns entrevistados, a depender da situação, o cliente se torna um problema por não ter certeza de quais são seus requisitos, o que ocasiona mudanças constantes no projeto, mais custos e retrabalhos.

Quadro 03 - Flexibilidade com relação ao cliente

Elementos/Casos	ALFA	BETA	DELTA	GAMA	ZETA
Nível de Participação do cliente no processo de desenvolvimento	Intermediário, visto que não consegue a participação em todos os casos.	Participação intensa dos clientes em várias etapas do projeto.	Intermediário, visto que não consegue a participação em todos os casos.	Participação intensa do clientes em várias etapas do projeto.	Participação intensa dos clientes em várias etapas do projeto.
Aspectos positivos e negativos da participação do cliente no processo	Positivo: Essencial, a fim de dividir responsabilidades e antecipar problemas no desenvolvimento de <i>software</i> . Negativo: Não sabe o que realmente necessita; Não compreende a importância de seguir o processo para o sucesso do projeto	Positivo: Essencial para o projeto, com o objetivo de responder rapidamente às mudanças que surgem. Negativo: Não sabe o que realmente necessita.	Positivo: Essencial para o projeto, com o objetivo levantar as informações e posteriormente validar o <i>software</i> . Negativo: Não sabe o que realmente necessita	Positivo: Essencial para o projeto, com o objetivo de responder rapidamente às mudanças que surgem, compartilhando conhecimento. Negativo: Não sabe o que realmente necessita; Não compreende a importância do seguir o processo para o sucesso do projeto.	Positivo: Essencial para o projeto a fim de dirimir questões relativas ao projetos, homologação e feedback. Negativo: não se importa muito com a participação no processo e preocupando-se estritamente com os prazos de entrega do sistema.
Atendimento das demandas de mercado	Atende, mas com dificuldades em relação a prazo e definição de escopo de projeto, que são constantes, ocasionando retrabalho e falta de alinhamento com a evolução tecnológica.	A entrega de partes do <i>software</i> é constante, contudo, o cliente muitas vezes solicita as demandas de maneira errada ocasionando retrabalho.	Atende em alguns casos, visto que quando o cliente não participa na coleta de informações, muitos projetos deixam de ser desenvolvidos.	A entrega de partes do <i>software</i> é constante, contudo, o cliente muitas vezes solicita as demandas de maneira errada.	Atende mas apresenta problemas com relação ao prazo, contudo, ressalta que a padronização dos processos traz confiabilidade nos produtos desenvolvidos.

Fonte: Pesquisa de Campo (2013)

Nesse sentido, os gestores de projetos dos processos Beta e Delta perceberam o cliente como um problema no desenvolvimento de *software* por conta desta incerteza sobre suas necessidades. Já os gerentes dos casos processos Alfa e Gama indicam que suas dificuldades em relação à lida com o cliente consistem na dificuldade em manter os clientes participando em todas as etapas do processo.

Quanto ao acompanhamento das mudanças tecnológicas e exigências de mercado, todos os gestores declararam que buscam absorver as novidades no setor. No entanto, os gestores responsáveis pelos casos Delta e Zeta ressaltaram que só alteram projetos em andamento caso a novidade tecnológica seja necessária ao projeto, buscando não ocasionar alterações no prazo de entrega. Caso não, o projeto é finalizado e, posteriormente, outro projeto é criado para realizar estas atualizações. Já o gerente que atua no caso Alfa disse realizar estas atualizações tecnológicas, mas que estas mudanças causam atrasos nos projetos.

Vale destacar também que os gerentes dos casos Beta e Gama buscam sempre acompanhar estas novidades por exigência dos clientes, como também por vantagem competitiva no mercado. Eles afirmaram fazer estas alterações facilmente, e atribuem esta facilidade ao processo adotado, que é baseado na metodologia ágil.

É possível observar que, pelo fato dos entrevistados dos casos dos processos Beta e Gama produzirem seus produtos para o mercado, existe a necessidade que estejam mais alinhados às necessidades dos clientes e às mudanças tecnológicas, a fim de lançar no mercado produtos inovadores. Nesse sentido, a flexibilidade no desenvolvimento de *software* é caracterizada como uma vantagem competitiva para esses dois casos (SETHI; SETHI, 1990; GIACHETTI *et al*, 2003; SCHMIDT, 2006; TRKMAN, 2010).

4.3 Flexibilidade no grupo de trabalho

Para delimitar os objetivos deste estudo considerou-se que a flexibilidade no grupo de trabalho refere-se à habilidade da equipe de trabalho em ajustar suas atividades de acordo com as condições de mudança, de modo que estes ajustes não resultem em uma desorganização do grupo (SETHI; SETHI, 1990; BENG, 2009).

Quadro 04 - Flexibilidade no grupo de trabalho

Elementos/Casos	ALFA	BETA	DELTA	GAMA	ZETA
<i>Flexibilidade Funcional</i>	Os funcionários tem seus papéis e atividades bem definidos, contudo, em casos críticos de urgência, pode haver o desvio de função temporário.	Não existe uma definição rígida de papéis e responsabilidades.	Os funcionários tem seus papéis e atividades bem definidos, contudo, quando ocorrem imprevistos, o remanejamento é utilizado como medida de contingência.	Não existe uma definição rígida de papéis e responsabilidades.	Os funcionários tem seus papéis e atividades bem definidos, contudo, é comum o desvio temporário.
<i>Capacidade Gerencial dos funcionários</i>	A equipe não tem liberdade de gerenciar suas prioridades e atividades.	A equipe tem liberdade de gerenciar suas atividades, e definir em conjunto suas tarefas para atender à entrega do produto.	A equipe não tem liberdade de gerenciar suas prioridades e atividades. Ficam focados somente nos aspectos técnicos.	A equipe tem liberdade de gerenciar suas atividades e definir em conjunto suas tarefas para atender a entrega do produto.	A equipe não tem liberdade de gerenciar suas prioridades e atividades.
<i>Autonomia das Decisões</i>	Centralizadas no gerente de projetos.	Decisão tomada em comum acordo entre os envolvidos no projeto.	Centralizadas no gerente de projetos.	Decisão tomada em comum acordo entre os envolvidos no projeto.	Centralizadas no gerente de projetos.
<i>Comunicação Interna</i>	Todos estão cientes das atividades que estão sendo executadas, por meio de treinamentos workshops e reuniões.	Todos estão cientes das atividades que estão sendo executadas, por meio de reuniões diárias, semanais e a cada término de projeto, em que são discutidas as lições aprendidas.	Somente os envolvidos na execução do projeto estão cientes do que está sendo feito, por meio de reuniões.	Todos estão cientes das atividades que estão sendo executadas, por meio de reuniões diárias, semanais e a cada término de projeto, em que são discutidas as lições aprendidas.	Todos estão cientes das atividades que estão sendo executadas, por meio de treinamentos workshops e reuniões.
<i>Reação da equipe diante de mudanças</i>	O grupo de trabalho não reage bem à mudança, sendo assim, buscam formas de se antecipar às mudanças.	Não reagem bem às mudanças, gerando desconforto na equipe quanto às alterações nos produtos, planejamento e prazos.	A equipe reage bem por considerar a mudança como um novo projeto que terá que entrar novamente na fila de prioridades.	A equipe não reage bem, principalmente porque estimam o prazo criando uma zona de conforto.	O grupo de trabalho não reage bem à mudança, muitos casos o projeto chega a ser descontinuado.

Fonte: Pesquisa de Campo (2013)

No que concerne à flexibilidade funcional, observou-se no Quadro 04, nos gestores dos processos Beta e Gama, que adotam metodologias ágeis, que todos os membros da equipe podem realizar qualquer uma das atividades do projeto. No restante dos casos, seus gestores dizem que há essa flexibilidade de funções, mas que somente é utilizada quando há necessidade por conta de algum aspecto contingencial.

Esta flexibilidade do funcionário é importante quando há a necessidade de responder às mudanças no projeto (LI; JIANG; KLEIN, 2007; XIA; LEE, 2009; LI et al., 2010; GÜNSEL *et al.*, 2012). Esta qualidade também é importante ao considerar que o gerenciamento em equipes de desenvolvimento de *software* está se tornando cada vez mais horizontal, ao contrário do gerenciamento hierárquico tradicional (BYRD; TURNER, 2000), ou seja, em muitos momentos, os membros da equipe também desempenham funções gerenciais.

Em relação à capacidade gerencial dos funcionários, os gerentes dos casos Beta e Gama acreditam que proporcionar esta liberdade é positivo, embora há gestores (casos Delta e Zeta) que acreditam que a falta desta autonomia também é boa para a equipe. No caso Delta, o gestor acredita que este foco da equipe em questões exclusivamente técnicas aumenta a produtividade do grupo. O gestor do caso Zeta também acredita que proporcionar autonomia à equipe pode não ser positivo, uma vez que nem todos os membros têm responsabilidade suficiente para autogerir suas atividades. O gerente do caso Alfa não opinou se esta autonomia grupal seria ou não positiva, apenas declarou que nem a equipe nem ele, como gestor, tem liberdade de definir suas prioridades, pois estas questões são decididas na diretoria executiva do banco.

Já os entrevistados dos casos Beta e Gama acreditam que a autonomia de suas equipes é positiva para o desempenho, principalmente em resposta a mudanças, como para o comprometimento do grupo com as metas do projeto, contribuindo para o sucesso do processo. Para Li, Jiang e Klein (2007), quando o gestor proporciona esta liberdade na equipe, ele estimula o espírito de equipe e a comunicação entre os membros do grupo, o que impacta positivamente o desempenho do projeto.

Resultados semelhantes foram encontrados em um estudo de Günsel *et al.* (2012) que indicou que a autonomia do grupo de trabalho afeta positivamente o

sucesso do produto de *software* no mercado, a velocidade que este produto vai para o mercado e a funcionalidade do *software*. Estes resultados sugerem que a sinergia entre os membros da equipe podem reduzir a duração dos projetos, permitir que as deficiências sejam resolvidas rapidamente e aumentar a funcionalidade do produto de *software* resultante (GÜNSEL *et al.*, 2012).

Ainda para estes autores, o gestor deve buscar aumentar a flexibilidade na sua equipe. Ao promover a autonomia, selecionar membros com diversas áreas de especialização e proficiências, estimulando-os a exercer suas diversas habilidades, o gestor contribui para um melhor desempenho do grupo (GÜNSEL *et al.*, 2012).

No que concerne à autonomia das decisões, somente nos casos beta e gama, a equipe tem autonomia para definir suas prioridades nas atividades a serem executadas e sua auto-organização. Nos casos Alfa e Zeta, as definições de prioridade são externas à equipe. No caso Delta, estas definições são determinadas pelo gestor.

Ainda no Quadro 04, em relação à comunicação interna, os gerentes de projetos dos casos estudados declararam que todos os membros da equipe acompanham o andamento de todo o projeto, conhecendo suas etapas e sua evolução.

No tocante à reação das equipes diante de mudanças, em quatro dos cinco casos estudados as equipes ficam insatisfeitas com mudanças que ocorrem no decorrer do projeto, mesmo aquelas que adotam práticas ágeis (Beta e Gama). Fez-se esta consideração em relação aos gestores que adotam práticas ágeis de desenvolvimento de *software* por conta do caráter adaptativo deste grupo de metodologias, ou seja, elas foram elaboradas considerando o ambiente mutável da indústria de *software* (SOMMERVILLE, 2007).

O gestor do projeto do caso Beta ressaltou a importância de seu papel como gestor para contornar este tipo de situação de insatisfação frente às mudanças junto à equipe, visto que a motivação dos membros da equipe é uma das várias funções do gestor de projetos (XIA; LEE, 2009; PRESSMAN, 2011).

Uma exceção encontrada nos resultados quanto às mudanças no projeto é o processo Delta. O gestor ressaltou que a equipe percebe estas mudanças como naturais, e o processo adotado absorve estas alterações de forma rápida. Vale

destacar o caráter híbrido do processo de desenvolvimento de *software* adotado por este gestor.

Para Li *et al.* (2010), as capacidades de reação às mudanças contribuem para a flexibilidade do grupo de trabalho, que interfere na qualidade do produto de *software* desenvolvido.

4.4 Flexibilidade no processo de desenvolvimento de *software*

A flexibilidade no processo de desenvolvimento de *software* consiste na capacidade de adaptação do processo a mudanças, permitindo isenções e melhorias contínuas ao projeto ao facilitar que novas partes sejam adicionadas ou que partes antigas do produto sejam substituídas sem muitos custos e de forma rápida (SETHI; SETHI, 1990).

Em relação à capacidade de adaptação a mudanças, observou-se no Quadro 05 que todos os processos estudados estão adaptados às organizações. Segundo Maia, Freitas e Nunes (2004), pelo caráter genérico dos modelos de processos, é necessário que haja estas adaptações referentes à organização que foi implantado. Ainda para estes autores, estas adaptações são importantes para os processos de *software*, pois buscam aumentar o nível de automatização, ocasionando menores custos e tempo despendidos no processo, somado a uma maior aceitação e otimização no uso de tais processos.

Ainda no Quadro 05, verificou-se que em quatro dos casos analisados (Alfa, Beta, Delta e Gama), os modelos de processos adotados permitem mudanças durante a execução do projeto. Nos casos Beta e Gama, que adotam processos ágeis, os gestores destacam que os processos permitem estas mudanças de forma rápida.

Já o Alfa, que adota dois tipos de processos de desenvolvimento (tradicional e ágil), o entrevistado ressaltou que os processos permitem estas mudanças, em caráter excepcional, mas ocasionam retrabalhos. No caso Delta, que adota um modelo próprio e híbrido, o gestor destacou que a depender da extensão da mudança, ela se torna um novo projeto. No Zeta, que adota um processo tradicional, as mudanças sempre se tornam novos projetos.

Nesse sentido, Reis (2003) explica que os modelos de processos de *software* possuem características peculiares porque envolvem pessoas que estão desempenhando tarefas criativas, ou seja, não é possível prever todo o desenvolvimento, devendo ser construído aos poucos.

Quadro 05 - Flexibilidade nos processos de desenvolvimento de *software*

Elementos/Casos	ALFA	BETA	DELTA	GAMA	ZETA
Capacidade de adaptação a mudanças	O processo permite adaptações somente em casos de urgência, considerando algumas etapas do processo como essenciais e obrigatórias.	Existe total liberdade nas adaptações, sendo inerente à situação daquele momento da equipe e das características do projeto.	Permite adaptações em meio a execução, e até existem casos que o processo deixa de ser utilizado.	Os processos passam por mudanças constantes e a cada projeto passa por modificações e amadurecimento.	As adaptações surgem levando em consideração custo-benefício da mudança no processo e dos riscos que aquela adaptação pode trazer para a qualidade do produto.
Permissão de caminhos alternativos	Existe a possibilidade de que caminhos alternativos sejam criados em meio a execução e que etapas não obrigatórias deixem de ser seguidas.	As mudanças são constantes e diante das situações do dia-a-dia, rotas alternativas podem ser criadas.	Existe a possibilidade de que caminhos alternativos sejam criados em meio a execução e que etapas não obrigatórias deixem de ser seguidas.	As cerimônias são consideradas importantes e devem ser seguidas, mas existe total liberdade nas adaptações em meio à execução.	Permite que se percorra caminhos alternativos para a elaboração do produto de <i>software</i> , apesar de considerar importante seguir o processo
Capacidade de iniciar projetos sem as informações completas	Exige um nível mínimo de informação com relação a previsibilidade de todo escopo do projeto.	Entende que o <i>software</i> é imprevisível, não havendo grande esforço no planejamento, permitindo assim, que o projeto seja iniciado sem a informação completa.	O projeto não segue para a etapa de desenvolvimento se não houver as informações completas do sistema.	Os projetos são iniciados sem haver a informação completa.	Exige um nível mínimo de informação, com relação a previsibilidade de todo escopo do projeto.
Permissão de melhorias futuras no <i>software</i>	Existem sistemas antigos que não permitem isenções e melhorias, contudo, os projetos atuais são pensados considerando a necessidade de adicionar e/ou substituir novas partes no <i>software</i> .	Os produtos já são pensados para que seja possível acrescentar/substituir novas partes a ele de forma fácil e sem muitos custos.	Os produtos já são pensados para que seja possível acrescentar/substituir novas partes a ele de forma fácil.	Os produtos já são pensados para que seja possível acrescentar/substituir novas partes a ele de forma fácil.	Somente nos novos produtos e produtos pilotos, os projetos em antigos não aceitam inserções e melhorias.
Os prazos diante de mudanças no projeto	Existe uma grande dificuldade na estimativa de prazo, nesse sentido, grande parte dos projetos é entregue fora do prazo.	Os prazos são negociados constantemente a cada solicitação de mudança. As entregas são quinzenais e são negociadas em módulos.	Existe um grande esforço na documentação e previsão de <i>software</i> , muitos projetos são descontinuados ou nem são iniciados.	Existe um esforço em gerenciar a expectativa do cliente, visto que com sua participação as mudanças passam a ser constantes aumentando escopo, custo e prazo.	Existe atraso em virtude do esforço na documentação dos projetos, que é considerado burocrático mas importante em virtude das auditorias internas e traz facilidade para responder os feedbacks.

Fonte: Pesquisa de Campo (2013)

Em relação à permissão de caminhos alternativos nos processos de desenvolvimento, todos os gestores declararam que os processos adotados permitem que etapas sejam modificadas ou não realizadas, de forma que em cada projeto é selecionado o caminho de execução mais apropriado, conforme recomendado por Schonenberg et al. (2007). Nesta direção, Reis (2003) indica que processos flexíveis devem permitir a escolha de caminhos alternativos buscando lidar melhor com a imprevisibilidade do desenvolvimento de *software*.

No entanto, vale destacar que todos os entrevistados destacaram que algumas etapas são imprescindíveis, a exemplo da documentação nos casos Delta e Zeta. Mesmo em um caso que adota metodologias ágeis, como o Beta, existem etapas essenciais, como as cerimônias do processo *Scrum*.

Nesse sentido, Moraes e Laurindo (2013) buscaram discutir a relação entre maturidade em gestão de projetos e o desempenho de projetos de TI e encontrou como resultado que quanto mais padronizado os processos, maior a maturidade na gestão de projetos de TI, o que aumenta o desempenho dos projetos.

Estas posturas dos gestores estão de acordo com Liu et al. (2008), que defendem a padronização do processo de desenvolvimento, com algumas etapas flexíveis. Para estes autores, a flexibilidade deve funcionar somente como mediadora entre a padronização e a melhoria de desempenho do processo.

No tocante à necessidade de se ter todas as informações sobre o sistema para iniciar o projeto, os gerentes dos casos Alfa, Beta, Gama e Zeta destacaram que iniciam seus projetos sem que todas as informações estejam especificadas, exigindo uma quantidade mínima de informações. Nesse sentido, Reis (2003) indica que processos que apresentam características flexíveis devem construir seus projetos aos poucos, lidando com informações incompletas.

No caso processo Delta, os projetos só são iniciados com todas as informações necessárias. Assim, se o cliente não fornecer ou não saber especificar todas as informações, a equipe, baseada em sua experiência de desenvolvimento e em produtos existentes no mercado, completa os dados necessários para o início do projeto.

Quanto à permissão de melhorias futuras no *software*, no Quadro 05, todos os casos declararam que os sistemas já são elaborados de forma que possibilite a adição e/ou substituição de novas partes. No entanto, os entrevistados dos casos

Alfa e Zeta, que trabalham em uma instituição financeira, ressaltam que ainda existem *softwares* antigos em uso que não permitem alterações.

Esta flexibilidade em relação à alteração do produto de *software* é importante para que a empresa acompanhe as mudanças tecnológicas do mercado de forma rápida e sem muitos custos, como também para que lance no mercado produtos inovadores de forma rápida (NADIR; NADARAJAH, 2012).

A partir dos dados, pode-se inferir que todos os gestores perceberam características flexíveis em seus processos de desenvolvimento de *software*. Ao considerar que a flexibilidade nos processos consiste na capacidade de se adaptar de forma criativa a mudanças, permitir isenções e melhorias contínuas nos produtos (KADIR; NADARAJAH, 2012), todos os gestores declararam que os processos investigados apresentam mais de uma destas características.

No que concerne aos prazos diante das mudanças no projeto, todos os casos apresentam alterações de prazos nos projetos, a depender da extensão da mudança. Nos casos Beta e Gama, que adotam práticas ágeis e têm entregas frequentes durante o projeto, os gestores declararam que negociam estes prazos a cada solicitação do cliente. Já os processos Alfa, Delta e Zeta, a depender da extensão da mudança, estas não são realizadas, a fim de evitar atrasos em relação ao prazo de entrega.

5 CONSIDERAÇÕES FINAIS

A pesquisa buscou investigar como os gestores de projetos de *software* percebem as características de flexibilidade no desenvolvimento de *software* no que concerne aos aspectos de flexibilidade com relação ao cliente, no grupo de trabalho e no processo de desenvolvimento, partindo da compreensão que os sistemas de informação são compostos por entidades técnicas e não técnicas.

Concluiu-se que as empresas que são fábricas de *software* (casos Beta e Gama) adotam processos ágeis, com menores ciclos de vida devido ao tamanho de seus projetos, e seguem uma tendência de acompanhar as mudanças tecnológicas que afetam o mercado de *software*, uma vez que isto se torna fundamental para seu posicionamento no mercado. Os entrevistados julgam a flexibilidade no

desenvolvimento de *software* importante e a consideram uma vantagem competitiva para empresa.

No tocante às empresas que não têm a produção de *software* como sua atividade fim (processos Alfa, Delta e Zeta), elas tendem a adotar modelos de processos tradicionais de desenvolvimento de *software*, com maiores ciclos de vida devido ao tamanho dos projetos. Os entrevistados buscam acompanhar as mudanças tecnológicas do setor, no entanto, estas atualizações comumente se tornam novos projetos. Os entrevistados consideram a flexibilidade no desenvolvimento de *software* importante, no entanto, os achados desta pesquisa evidenciaram que, por questões de ordem organizacional, muitas medidas flexíveis não são possíveis de serem adotadas.

Esta pesquisa não pretendeu esgotar o debate sobre o objeto de estudo, mas sim discutir aspectos pouco estudados dentro da temática engenharia de *software*. Assim, a pesquisa contribuiu para o campo de estudos da área ao ampliar o conhecimento a respeito do desenvolvimento de *software* em Sergipe, ainda pouco explorado na literatura, pois ressaltou as principais características e dificuldades enfrentadas pelos gestores de projetos. Dessa forma, acredita-se ter contribuído para as pesquisas acadêmicas na área, assim como para os gestores de projetos de desenvolvimento de *software* por trazer um tema importante para o sucesso de empresas da área.

Destarte, espera-se que seja possível que a partir desta pesquisa surjam outros estudos que tenham um enfoque semelhante, e que possam contemplar as particularidades do desenvolvimento de *software* no estado e no país.

Nota:

Artigo fruto de dissertação de mestrado.

REFERÊNCIAS

ABBAS, N.; GRAVELL, A. WILLS, G. An Empirical Comparison of Two Agile Projects in the Same Organization. IN: **Agile 2008 Conference**, Toronto, Canada. 2008.

ABRAHAMSSON, P.; WARSTA, J.; SIPONEN, M. T.; RONKAINEN, J. New directions on agile methods: a comparative analysis. **International Conference on Software Engineering**, vol. 5, n. 3, pp. 244-254, 2003.

- BARDIN, L. **Análise de conteúdo**. Lisboa, 2008.
- BASILI, V.R. **Software modeling and measurement**: the goal/question/metric paradigm. Technical report, CS-TR- 2956, Department of Computer Science, University of Maryland, College park, MD 20742, September, 1992.
- BECK, K. **Extreme programming explained**: Embrace change. Boston: Addison-Wesley, 2000.
- BECK, K.; BEEDLE, M.; BENNEKUM, A.; COCKBURN, A.; CUNNINGHAM, W.; FOWLER, M.; GRENNING, J.; HIGHSMITH, J.; HUNT, A.; JEFFRIES, R.; KERN, J.; MARICK, B.; MARTIN, R.; MELLOR, S.; SCHWABER, K.; SUTHERLAND, J.; THOMAS, D. **Manifesto for Agile Software Development**, 2001. Disponível em: <<http://www.agilemanifesto.org/>>. Acesso em: 20 ago. 2012.
- BEYNON-DAVIES, P.; CARNE, C.; MACKAY, H.; TUDHOPE, D. Rapid application development (RAD): an empirical review. **Eur. J. Inf. Syst.**, vol. 8, n. 3, pp. 211–223. 1999 <http://dx.doi.org/10.1057/palgrave.ejis.3000325>
- BENG, J.M. Supply Chain Flexibility aspects and their impact on customer satisfaction. Dissertação (Mestrado em Supply Chain Management) – **Faculty of Management**, Open University the Netherlands. 2009.
- BOEHM, B. Get ready for agile methods, with care. **Computer**, vol. 35, n. 1, pp. 64-69, 2002. <http://dx.doi.org/10.1109/2.976920>
- BOEHM, B. A spiral model of *software* development and enhancement. **Computer**, vol. 21, n. 5, pp. 61–72. 1988. <http://dx.doi.org/10.1109/2.59>
- BYRD, T.A.; TURNER, D.E. Measuring the Flexibility of Information Infrastructure: Exploratory Analysis of a Construct. **Journal of Management Information Systems**. 17(1). pp. 167–208, 2008.
- CAVALCANTI, M.; GOMES, E. A sociedade do conhecimento e a política industrial brasileira. Brasília: **MDIC**, 2001. Disponível em: <<http://portal.crie.coppe.ufrj.br/portal/data/documents>>. Acesso em: 31 dez. 2012.
- CHERRY, S.; ROBILLARD, P. N. The social side of *software* engineering - A real ad hoc collaboration network. **International Journal of Human-Computer Studies**, vol. 66, n. 7, pp. 495-505, 2008. <http://dx.doi.org/10.1016/j.ijhcs.2008.01.002>
- CHESBROUGH, H.W. **Inovação Aberta**: Como Criar e lucrar com a tecnologia. Porto Alegre: Bookman, 2012.
- CHOU, M.; TEO, C-P.; ZHENG, H. **Process Flexibility**: Design, Evaluation and Applications. 2008. Disponível em: <http://bschool.nus.edu/Staff/bizteocp/flexireview.pdf>. Acessado em: 13 dez 2012.
- CRESWELL, J. W. **Research Design**: qualitative, quantitative and mixed methods approaches. Thousand Oaks: Sage. pp. 179- 204, 2003.

DINGSØYR, T.;SRIDHAR, S.;BALIJEPALLY, V.;MOE, N. B. A Decade of Agile Methodologies: Towards Explaining Agile *Software* Development. **Journal of Systems and Software**, vol. 85, pp. 1213-1221, 2012. <http://dx.doi.org/10.1016/j.jss.2012.02.033>

EDEN, A. H.; MENS, T. Measuring *Software* Flexibility. **IEE Software**, 153(3), pp. 113–126, 2006. <http://dx.doi.org/10.1049/ip-sen:20050045>

GIACHETTI, R.E.; MARTINEZ, L.D.; SÁENZ, O.A.; CHEN, C.S. Analysis of the structural measures of flexibility and agility using a measurement theoretical framework. **Int. J. Production Economics**, vol. 86, pp. 47-62, 2003. [http://dx.doi.org/10.1016/S0925-5273\(03\)00004-5](http://dx.doi.org/10.1016/S0925-5273(03)00004-5)

GUNSEL, A.; AÇIKGOZ, A; TUKEL, A.; OGUT, E. The Role Of Flexibility On *Software* Development Performance: An Empirical Study On *Software* Development Teams. **Procedia - Social and Behavioral Sciences**, vol. 58, pp. 853-860, 2012.

HIGHSMITH, J, & COCKBURN, A. Agile *software* development: the business of innovation. **Computer**, vol. 34, pp. 120-127, 2001. <http://dx.doi.org/10.1109/2.947100>

LAN, C.; RAMESH, B. Agile requirements engineering practices: an empirical study. **IEEE Software**, January/February, pp. 60-67. 2008.

LARMAN, C.; BASILI, V. R. **Iterative and incremental developments**: a brief history. **Computer**, 2003. <http://dx.doi.org/10.1109/MC.2003.1204375>

LEE, G; XIA, W. Toward agile: an integrated analysis of Quantitative and qualitative field data On *software* development agility. **MIS Quarterly**, V. 34 N. 1, pp. 87-11. 2010.

LI, Y.; CHANG, K-C; CHEN, H.G; JIANG, J.J. *Software* development team flexibility antecedents. **The Journal of Systems and Software**, vol. 83, pp. 1726-1734, 2010. <http://dx.doi.org/10.1016/j.jss.2010.04.077>

LI, Y.; JIANG, J.; KLEIN, G. The Antecedents of IS *Software* Development Team Flexibility - **Research in Progress**. International Research Workshop on IT Project Management, 2007.

LIU, J.Y-C; CHEN, V.J.; CHAN, C-L; LIE, T. The impact of *software* process standardization on *software* flexibility and project management performance: Control theory perspective. **Information and Software Technology**, vol, 50, pp. 889-896, 2008. <http://dx.doi.org/10.1016/j.infsof.2008.01.002>

MAIA, A.B.; FREITAS, A.V.P.; NUNES, D.J. Um Modelo para Auxiliar a Adaptação de Processos de *Software*. IN: **Anais do IV Congresso Brasileiro de Computação**, p. 155-160, 2004.

MOE, N. B.; DINGSØYR, T.; DYBÅ, T. A teamwork model for understanding an agile team: A case study of a Scrum project. **Information and Software Technology**, vol. 52, n. 5, pp. 480-491, 2010. <http://dx.doi.org/10.1016/j.infsof.2009.11.004>

MORAES, R.O.; LAURINDO, F.J.B. Relações entre o desempenho dos projetos de TI e a maturidade em gestão de projetos. **Revista Produção Online**, 13(1), 2013. P. 61-83. <http://dx.doi.org/10.14488/1676-1901.v13i1.1027>

PRESSMAN, R. S. **Engenharia de software**. 6. ed. São Paulo: McGraw-Hill, 2006.
SCHMIDT, R. Flexibility in Service Processes. CAISE'06 Workshop on Business Process Modelling, **Development, and Support** (BPMDS'06). Luxemburg. June, pp. 5-9. 2006.

SCHWABER, K. Agile Project Management with Scrum. **Microsoft Press**, p. 192, 2004.

SETHI, A.K.; SETHI, S.P. Flexibility in manufacturing: A survey. **The International Journal of Flexible Manufacturing Systems**, vol. 2, pp. 289-328, 1990. <http://dx.doi.org/10.1007/BF00186471>

SIDKY, A.S. ARTHUR, J.D. BOHNER, S.A. A disciplined approach to adopting agile practices: the agile adoption framework. **ISSE**, vol. 3, n. 3, pp. 203-216. 2007.

REIS, C.A.L. **Uma abordagem flexível para execução de processos de software evolutivos**. Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio Grande do Sul. 2003.

ROYCE, W. W. Managing The Development Of Large *Software* Systems. **IEEE WESCON**, (August), pp. 1-9, 1970.

SCHONENBERG, M.H.; MANS, R.S.; RUSSEL, N.C.; MULYAR, N.A.; AALST, W.M.P. **Towards a Taxonomy of Process Flexibility** (Extended Version). 2007.

SOARES, M. D. S. Comparação entre Metodologias Ágeis e Tradicionais para o Desenvolvimento de *Software*. **Journal of Computer Science**, vol. 27, n. 2, p. 6, 2003.

SOMMERVILLE, I. **Engenharia de software**. São Paulo: Pearson, 2007.

TRKMAN, P. The Critical Success Factors of Business Process Management. **International Journal of Information Management**, vol. 30, n. 2, pp. 125-134, 2010. <http://dx.doi.org/10.1016/j.ijinfomgt.2009.07.003>

VINEKAR, V. Towards A Theoretical Framework Of Information Systems Development Strategy: **The Contingent Effects Of Organizational**, pp. 181-186, 2006.

WILLIAMS, L; COCKBURN, A. Agile *software* development: It's about feedback and change. **Computer**, vol. 36, pp. 39-43, 2003.
<http://dx.doi.org/10.1109/MC.2003.1204373>



Artigo recebido em 18/08/2013 e aceito para publicação em 14/04/2014
DOI: <http://dx.doi.org/10.14488/1676-1901.v14i3.1658>