

## Avaliação dos métodos de produção de software utilizados no desenvolvimento de aplicações periféricas aos ERP's

Valter Yogui

Centro Paula Souza – SP – Brasil

[valter.yogui@fatecsp.br](mailto:valter.yogui@fatecsp.br)

Marilia Macorin de Azevedo

Centro Paula Souza – SP – Brasil

[marilia.azevedo@fatec.sp.gov.br](mailto:marilia.azevedo@fatec.sp.gov.br)

**Resumo** - O objetivo deste trabalho é avaliar quais os métodos de produção de software são utilizados na confecção de soluções, sub-sistemas, aplicativos ou programas na periferia dos ERP's (*Enterprise Resource Planning*), para suprir as necessidades de negócio das empresas e que não são cobertas por estes tipos de sistemas integrados. A partir de pesquisa quantitativa com implementadores de produtos ERP's de mercado, foram identificados os métodos empregados para o desenvolvimento destas soluções periféricas, denominadas também como *customizações*, e analisadas as suas aderências aos métodos tradicionais de Engenharia de *Software*, aos Métodos Ágeis, ou aos modelos de produção adaptados.

**Palavras-chave:** erp, customização, engenharia de software, método ágil

**Abstract** - The aim of this work is evaluate which methods of software development is used in the manufacture of solutions, sub-systems, applications or programs surround ERP's (*Enterprise Resource Planning*), to supply business needs of companies which are not covered by these types of integrated systems. From quantitative research on ERP deployment professionals, identified the methods used to develop these peripheral solutions, also known as *customizations*, and analyzed their adherence to traditional methods of Software Engineering, Agile Methods or adapted production models.

**Keywords:** erp, customization, software engineering, agile method

### 1 . Introdução

Ao optarem pela utilização de um ERP (*Enterprise Resource Planning*), as empresas têm o benefício da integração dos sistemas, obtendo inclusive, conforme Monsore (2009), uma redução nos custos que antes existiam com a manutenção de sistemas próprios. Com a utilização do ERP, as demais aplicações necessárias são desenvolvidas na sua periferia, sendo denominadas *customizações*.

Segundo Bassi Filho (2008), a demanda por softwares aumenta enquanto os prazos para o desenvolvimento dos mesmos diminuem. As empresas buscam formas de desenvolver seus softwares procurando obter entregas mais rápidas e agregar valor para o usuário final. No desenvolvimento de aplicações periféricas aos ERP's, os próprios

fabricantes estipulam e oferecem um método próprio para a produção destas aplicações, sem obedecer aos critérios formais da Engenharia de Software tradicional.

O objetivo deste trabalho é analisar que métodos de produção de software são utilizados para o desenvolvimento destas soluções periféricas ou *customizações*, e a sua semelhança à Engenharia de Software tradicional, aos Métodos Ágeis, ou se são modelos de produção adaptados dos mesmos.

## 2. Referencial Teórico

Segundo Laudon e Laudon (2004), ERP (*Enterprise Resource Planning*) são sistemas integrados de informações que tratam de todos os dados e processos de uma organização em um único e abrangente sistema. O objetivo é gerir de forma centralizada todos os dados relevantes das atividades da empresa ou organização para permitir que possam ser analisados posteriormente e transformados em informações úteis.

Antes do surgimento do conceito de ERP, cada departamento ou setor das empresas possuía um software ou sistema específico para suas necessidades, atendendo perfeitamente o seu foco, mas sem nenhum tipo de interação entre os diversos sistemas das empresas. O principal benefício trazido pelo ERP foi a integração destes sistemas, quando a informação passou a ser lançada uma única vez, ficando disponível logo em seguida para todas as áreas da empresa. Novas funções ou necessidades são preenchidas pela instalação de pacotes mais abrangentes proporcionados pelos próprios fornecedores de ERP, ou pela instalação de softwares complementares ou periféricos (DAVENPORT, 2002).

O objetivo da Engenharia de Software é gerar valor e contribuir para o sucesso das empresas através do desenvolvimento de sistemas de informação. As abordagens metodológicas desenvolvidas nesta direção podem ser classificadas como métodos rígidos ou ágeis, e que têm em comum o mesmo conjunto de premissas em sua proposta principal. Conforme Sommerville (2007) e Pressman (2006), temos o escopo bem definido, ou seja, deve-se estabelecer o problema a ser resolvido e as necessidades a serem atendidas, dentro das condições determinantes para um atendimento satisfatório; a especificação do sistema, que corresponde a descrição das funções do software, de modo a implementar a solução do problema, o cumprimento e o atendimento das necessidades, e que deve ser uma forma detalhada para permitir o desenho do sistema, sua construção e respectivo teste, da forma mais fidedigna possível; e a relação existente entre as duas definições anteriores, onde a complexidade do escopo determina os prazos e os custos do desenho do sistema, sua construção e seus testes.

Sobre os métodos tradicionais de Engenharia de Software, aqui denominados métodos rígidos, apresentamos 2 (dois): o método Cascata e o método Espiral. O método Cascata foi inicialmente definido por Royce (1970) e amplamente utilizado dentro da Engenharia de Software, com uma abordagem pragmática e linear que a indústria de software usou para a construção de todos os tipos de sistemas. Neste método, o software final é obtido através da execução de etapas sistematicamente definidas.

Um conceito que permanece vigente há pelo menos vinte anos, definido por Pressman (1995) e revalidado em Pressman (2006), o método Cascata é um processo que segue linearmente as etapas de engenharia do sistema, análise de requisitos, projeto, geração de código, testes e manutenção. Em cada uma destas etapas, um conjunto pré-estabelecido de atividades é realizado de forma que os artefatos produzidos em cada etapa sirvam de entrada para a etapa seguinte. As atividades de cada etapa são: na Engenharia do sistema, a coleta e a análise de requisitos e dos equipamentos para definir as necessidades tecnológicas e as possíveis limitações; na Análise de Requisitos, a

identificação das funções desejadas para o software e a aquisição de conhecimento detalhado sobre o domínio de negócios que o sistema propõe tratar, identificando os tipos de interação, as necessidades de interface com outros sistemas e todos os fatores que determinam o modo como os usuários e outros sistemas utilizarão o novo software; no Projeto, a definição das características da implementação, as soluções de arquitetura e modelagem do sistema, as estruturas de dados e as interfaces entre os elementos do sistema, de modo a cobrir todos os requisitos identificados nas etapas anteriores, sendo que o resultado final desta etapa é a especificação detalhada do código a ser produzido, definindo como cada parte do sistema deverá ser implementada; na Geração do código, a produção do software conforme descrito na especificação do projeto; nos Testes, os elementos e componentes do sistema são validados de acordo com a sua especificação; na Manutenção, a definição e geração de uma nova versão do software, a partir da atual, onde poderão ser feitos ajustes necessários. Com a versão atual em produção, a nova versão do software volta para alguma das etapas anteriores para a inclusão de novas funcionalidades ou para ajustes ou melhorias.

O modelo Espiral, proposto inicialmente por Boehm (1986) e revalidado por Hansen (2000) e Boehm (2000) apresenta duas diferenças em relação ao modelo Cascata: a primeira é a introdução de um modelo iterativo, onde as etapas do desenvolvimento são realizadas várias vezes no formato de ciclos; e a segunda é a inclusão de uma etapa de Análise de riscos, quando os envolvidos no projeto podem avaliar e tomar as decisões conforme as dificuldades emergentes durante o desenvolvimento.

Cada iteração é formada por quatro etapas: Planejamento, Análise de riscos, Engenharia e Avaliação do cliente. O produto evolui a cada iteração, aproximando-se do que será a versão que entrará em produção. As atividades de cada etapa são: no Planejamento são coletados os requisitos principais para realizar um planejamento e determinar os objetivos e as restrições; na Análise de riscos, os riscos associados ao projeto são identificados através do levantamento das incertezas associadas ou oriundas dos requisitos, e a partir da análise dos principais riscos procuram-se soluções ou alternativas que reduzam as incertezas identificadas, sendo que, diante deste cenário, pode-se decidir pela continuidade ou interrupção do projeto até uma melhor definição; na Engenharia, o produto definido na etapa de Planejamento é construído, produzido por partes e através de várias iterações do ciclo de desenvolvimento; na Avaliação do cliente, o cliente analisa e avalia o trabalho realizado na etapa da Engenharia de acordo com os objetivos definidos no Planejamento, acrescenta seus comentários, sugestões, modificações e críticas, as quais serão avaliadas para a próxima iteração.

Os Métodos Ágeis de desenvolvimento de software, segundo Melo e Ferreira (2010), vêm ganhando crescente popularidade desde o início da última década. Estes métodos são regidos pelo Manifesto Ágil de Desenvolvimento de Software (BECK et al, 2001). Com o objetivo de produzir entregas mais rápidas e agregar valor para o cliente, as empresas utilizam métodos de desenvolvimento menos rígidos e mais ágeis. Segundo Carvalho e Mello (2009), a percepção dos usuários finais é que a produtividade na utilização do método ágil é um fator relevante. O aumento da produtividade e a redução de prazos no desenvolvimento de software utilizando Métodos Ágeis são diagnosticados também por Melo e Ferreira (2010). E no caso de empresas menores, segundo Dybå (2000), há uma necessidade de improvisação de métodos de desenvolvimento em oposição à rigidez e a formalização dos métodos tradicionais, que enfatizam mais o planejamento do que a antecipação de possíveis problemas no desenvolvimento.

A proposta dos Métodos Ágeis é eliminar gastos com documentação excessiva e burocrática, enfatizar a interação entre as pessoas e nas atividades que efetivamente trazem valor e produzir software com qualidade (BECK et al, 2001), obedecendo a 4 (quatro) premissas fundamentais: os Indivíduos e iterações são mais importantes do que

a obediência a processos e utilização de ferramentas; o Software funcionando é mais importante do que a existência de uma documentação completa e exata; a Colaboração com o cliente é mais importante do que a negociação de contratos e o tratamento de mudanças; e a Adaptação a mudanças é mais importante do que somente seguir o plano inicial especificado no contrato.

Para obter resultados práticos em um período menor do que os obtidos pelos métodos de desenvolvimento de software tradicional, a diretriz passou do processo para o produto, sendo que etapas do modelo tradicional foram modificadas ou dispensadas, resultando nas seguintes características: um Desenvolvimento iterativo, que acontece em ciclos (iterações), cujo objetivo é produzir e integrar partes do software, sendo que no fim de cada ciclo o software é entregue ao cliente e o restante do desenvolvimento é direcionado pelas diretrizes do seu *feedback*; nos Testes, a implementação e os testes acontecem conjuntamente; o Desenvolvimento é incremental durante as iterações; as Estimativas são utilizadas com transparência, admitindo a incerteza do valor estimado, que são diminuídas com a aquisição de mais conhecimento; na Negociação o custo é definido pelo tamanho da equipe de desenvolvimento; e a Priorização é baseada na adaptação a mudanças, com planejamento mais detalhado para o curto prazo.

Dentre os Métodos Ágeis destacam-se 2 (dois): o XP (Programação eXtrema) e o Scrum. Segundo o Beck (1999), as etapas do XP são: a Exploração, com o intuito de conhecer as necessidades e intenções do cliente, com produção de protótipos para avaliar com precisão as tecnologias em questão e comparar os resultados; o Planejamento, cujo objetivo é descrever as funcionalidades desejadas e estimar o esforço de trabalho em conjunto com os clientes; a Implementação, com a atividade de desenvolver objetivando concluir a iteração, com a aprovação da implementação pelos clientes através de testes funcionais e considerando as mudanças nas funcionalidades ou os novos fatores que impactam na velocidade de desenvolvimento para o planejamento da próxima iteração. São propostas 12 práticas para serem executadas no dia-a-dia do trabalho: a) versões pequenas; b) Jogo do planejamento; c) *Design* Simples; d) Programação em pares; e) Testes; f) Refatorações; g) Integração contínua; h) Propriedade coletiva do código; i) Ritmo sustentável; j) Cliente presente; k) Metáfora; e l) Padrões de código.

No método SCRUM, segundo Schwaber e Sutherland (2013) a equipe de desenvolvimento trabalha unida com o objetivo de entregar o software e tem autonomia para definir a tática para esta consecução. O Scrum atua principalmente na gerência do projeto, sem determinar como a equipe executará as tarefas de programação. Esta abordagem favorece a auto-organização da equipe e permite a integração com outros métodos ágeis que focam nas práticas de programação, como por exemplo, o próprio XP. Três papéis estão presentes no método: o Dono do produto deve possuir a visão do produto em vários níveis; a Equipe é uma unidade dentro do método e deve ser multi-funcional e auto-suficiente; e o ScrumMaster deve possuir conhecimento de todo o processo para garantir que ele seja seguido e deve manter uma ampla visão sobre o projeto. O Scrum caracteriza-se como um processo empírico e adaptativo, com as fases: Planejamento, Sprint (entrega de uma parte do produto) e Avaliação.

Os ERP's são considerados sistemas de informação integrados e padronizados, e qualquer alteração, modificação ou adição de funcionalidades periféricas, para cumprir características particulares ou específicas de uma empresa, precisa ser desenvolvida sob a ótica da Engenharia de Software. Analisados teoricamente os tipos de métodos existentes para produção de software, tanto rígidos como ágeis, torna-se interessante diagnosticar, com pesquisa de campo, qual a tendência de escolha e de utilização destes métodos para o desenvolvimento destas aplicações periféricas.

### 3. Método

Este estudo pode ser classificado como descritivo, pois apresenta pesquisa bibliográfica sobre os conceitos de ERP (*Enterprise Resource Planning*) e os tipos de métodos de produção de software que poderiam ser utilizados para o desenvolvimento de aplicações periféricas aos ERP's.

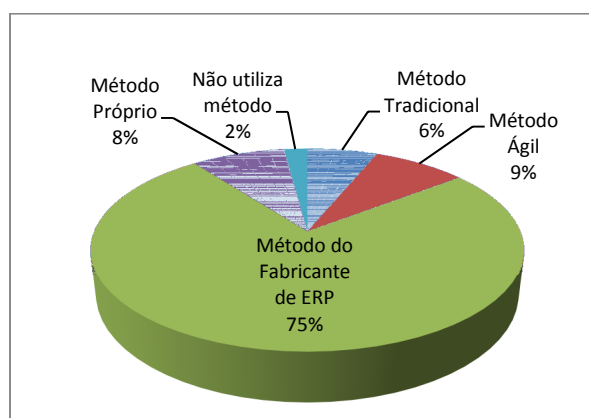
O método de pesquisa adotado foi o de levantamento de dados e pesquisa para a obtenção de dados de fontes reais, com interrogação direta dos sujeitos com questionário de perguntas com múltipla escolha e alternativa para descrição específica. A delimitação dos sujeitos foi determinada por uma amostra por conveniência do pesquisador.

A abordagem metodológica foi quantitativa, a partir de pesquisa que pode ser classificada como do tipo *survey*.

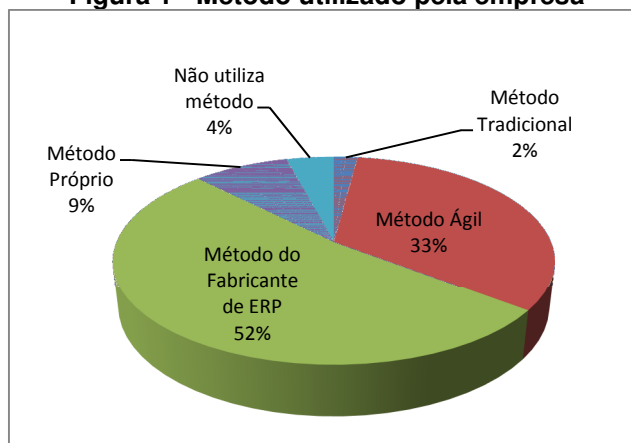
A população pesquisada foram profissionais gestores de projetos de diferentes empresas que implementam ERP na cidade de São Paulo. Foram selecionados 91 (noventa e um) gestores de projetos. A coleta de dados foi realizada por meio da aplicação de um questionário para os sujeitos pesquisados, que foi enviado por email direta e individualmente pelo pesquisador. O questionário baseou-se em perguntas para identificar qual o método de produção de software existente e adotado pela empresa, qual o método considerado mais adequado pelo profissional e qual a respectiva descrição das etapas deste método. Optou-se por realizar a pesquisa desta forma devido à facilidade e o menor trabalho para que os usuários pudessem responder de forma rápida e eficaz. A pesquisa foi realizada durante os dias 07/07/2014 a 15/07/2014. Foram recebidos 55 (cinquenta e cinco) questionários respondidos, sendo que 4 (quatro) foram desconsiderados por conter respostas com mais de 1 (uma) opção escolhida e outros 3 (três) foram desconsiderados porque correspondiam a profissionais que atualmente se encontram trabalhando fora da cidade de São Paulo, totalizando 48 (quarenta e oito) questionários válidos para análise neste trabalho.

### 4. Resultados e Discussão

Conforme apresentado na figura 1, o Método do fabricante de ERP apresentou-se como o método de produção de software mais utilizado pelas empresas no desenvolvimento de *customizações*, com o percentual de 75% de utilização. No entanto, conforme a figura 2, na apuração do método considerado mais adequado pelos profissionais para o desenvolvimento da mesma tarefa, apesar deste método apresentar-se também como o mais preferido, com índice de 52%, nota-se o aumento da preferência pelo Método Ágil, de 9% para 33%.

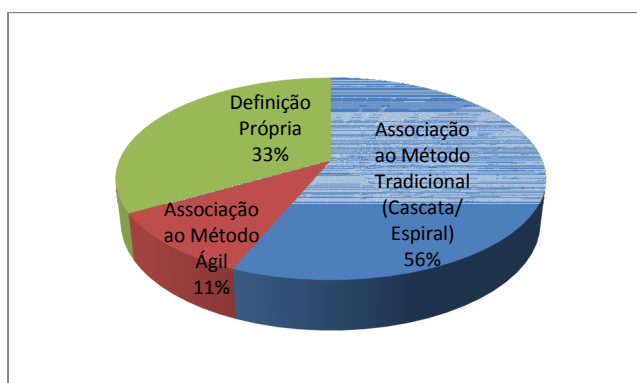


**Figura 1 - Método utilizado pela empresa**



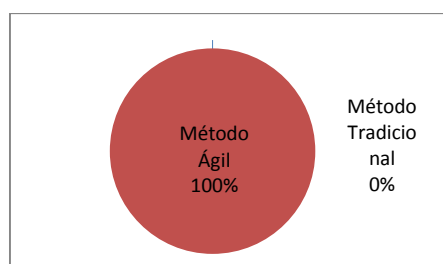
**Figura 2 - Método considerado mais adequado**

Conforme figura 3, independente do método utilizado na empresa, na exposição das descrições de etapas dos diferentes métodos, 56% dos profissionais optaram, como método mais adequado, pela associação às etapas do método tradicional (Cascata ou Espiral), 11% pela associação às etapas do Método Ágil e o restante optou pela própria definição das etapas, ou seja, definindo método próprio em 33%.

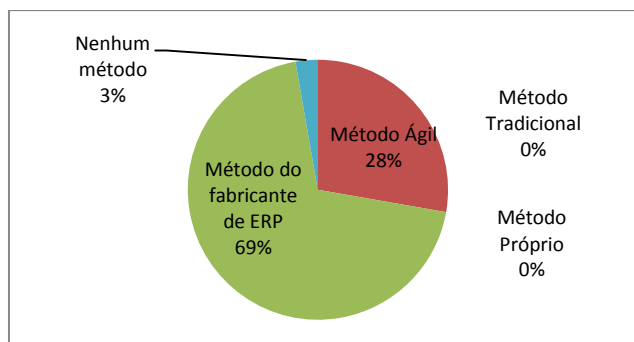


**Figura 3 - Escolha da sequência das etapas do Método considerado mais adequado**

Conforme figura 4, o Método Ágil, nas empresas que o utilizam, teve a preferência unânime dos profissionais como o método mais adequado para o desenvolvimento de *customizações*. No entanto, conforme figura 5, nas empresas que utilizam o Método do fabricante de ERP, não foi encontrada esta unanimidade na preferência dos profissionais, tendo o Método Ágil um índice de 28% de preferência, correspondente a 10 (dez) profissionais dentro do total de 36 (trinta e seis).

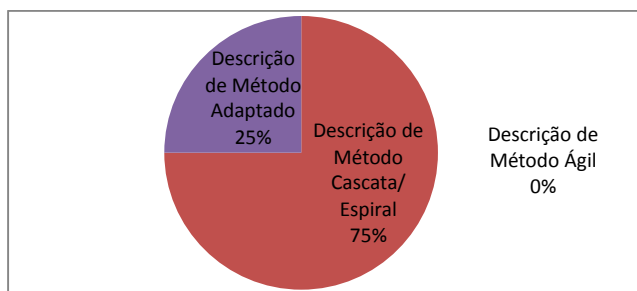


**Figura 4 - Nas empresas que utilizam o Método Ágil: Qual o Método considerado mais adequado?**

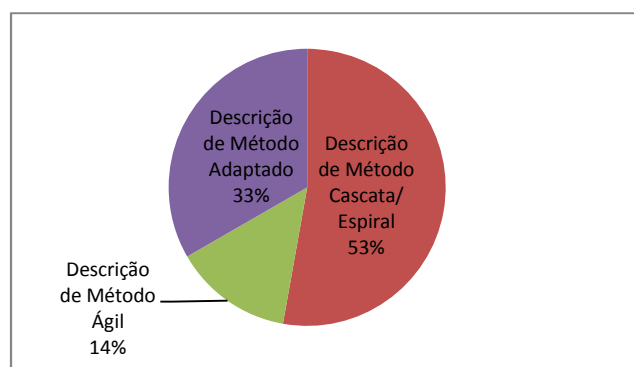


**Figura 5 - Nas empresas que utilizam o Método do fabricante de ERP: Qual o método considerado mais adequado?**

Nas empresas que utilizam o Método Ágil ou o Método do fabricante de ERP, conforme figuras 6 e 7, quando oferecidas somente as descrições de etapas dos diferentes métodos, a escolha do método mais adequado pela associação às etapas do método tradicional (Cascata ou Espiral) é predominante, com 75% e 53% respectivamente a cada método. Em compensação, a escolha do método mais adequado pela associação às etapas do Método Ágil é maior nas empresas que utilizam o Método do fabricante de ERP (14%) do que nas empresas que utilizam o método Ágil, que não existiu (0%).

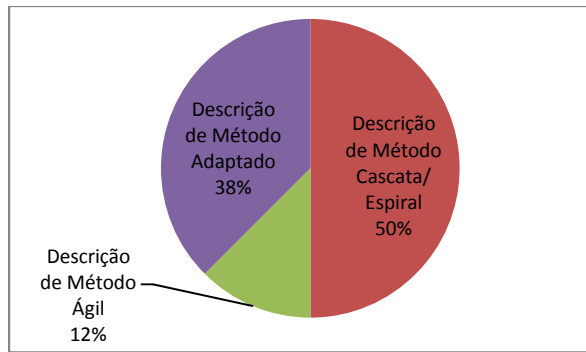


**Figura 6 - Nas empresas que utilizam o Método Ágil: Qual descrição de método é escolhida como mais adequada?**

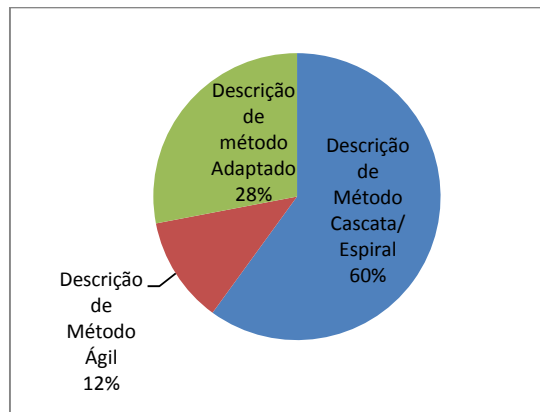


**Figura 7 - Nas empresas que utilizam o Método do fabricante de ERP: Qual descrição de método é escolhida como mais adequada?**

Quando o Método Ágil ou o Método do fabricante de ERP é considerado mais adequado, conforme figuras 8 e 9, e oferecidas somente as descrições de etapas dos diferentes métodos, a escolha do método mais adequado pela associação às etapas do método tradicional (Cascata ou Espiral) também é predominante, com 50% e 60% respectivamente a cada método. E a escolha do método mais adequado pela associação às etapas do Método Ágil foi igual nas empresas que consideram mais adequado o Método do fabricante de ERP como nas empresas que consideram o Método Ágil (12%).



**Figura 8 - Quando o Método Ágil é considerado mais adequado: Qual a descrição das etapas do Método?**



**Figura 9 - Quando o Método do fabricante de ERP é considerado mais adequado: Qual a descrição das etapas do Método?**

## 5. Considerações finais

O Método do fabricante de ERP apresentou-se como o método de produção de software mais utilizado pelas empresas no desenvolvimento de *customizações*, com o percentual de 75% de utilização (conforme figura 1).

Apresenta-se como método considerado mais adequado pelos profissionais, para o desenvolvimento de *customizações*, o método tradicional (Cascata ou Espiral). Tal fator pode ser identificado pela associação às etapas deste método: a) independente do método que é utilizado na empresa (56% conforme figura 3), b) nas empresas que utilizam o Método Ágil (75% conforme figura 6), c) nas empresas que utilizam o Método do fabricante de ERP (53% conforme figura 7), d) quando o Método Ágil é considerado mais adequado (50% conforme figura 8), e e) quando o Método do fabricante de ERP é considerado mais adequado (60% conforme figura 9).

Outro método considerado adequado pelos profissionais foi o Método Ágil: a) independente do método adotado pela empresa em que trabalham (33% conforme figura 2), b) nas empresas que utilizam o Método do fabricante de ERP (28% conforme figura 5), c) pela associação às etapas do Método Ágil nas empresas que utilizam o Método do fabricante de ERP (14% conforme figura 7), d) pela associação às etapas do Método Ágil nas empresas que consideram mais adequado o Método do fabricante de ERP (12% conforme figura 8), e e) pela associação às etapas do Método Ágil nas empresas que consideram mais adequado o Método Ágil (12% conforme figura 9).

Apesar do Método Ágil ser considerado adequado para 33% dos profissionais, (conforme figura 2) ressalta-se a possibilidade da falta de conhecimento dos princípios e características do mesmo por parte destes, visto que somente 11% (conforme figura 3)



associaram e reconheceram o método quando oferecida a descrição das etapas do mesmo. Tal possibilidade também se apresenta: a) quando, nas empresas que utilizam Método Ágil, não foi feita nenhuma escolha por associação à descrição das etapas do Método Ágil (0% conforme figura 6), b) quando o Método Ágil é considerado mais adequado e o reconhecimento da descrição das etapas do método é apenas 12% e não 100% (conforme figura 8).

Sendo assim, torna-se necessário que as empresas invistam em treinamento dos profissionais para a aquisição de conhecimento do Método Ágil, caso exista o intuito de utilizar o mesmo como outra possibilidade para desenvolvimento de *customizações*.

Como sugestão para a continuidade e o aprimoramento dos resultados deste estudo, esta pesquisa poderá futuramente ser complementada analisando-se as variações decorrentes do grau de experiência dos profissionais e de sua formação acadêmica ou técnica, uma segmentação por porte de empresas pesquisadas bem como uma análise de qual(is) método(s) teria(m) mais aderência aos Métodos dos fabricantes de ERP.

## Referências

BASSI FILHO, D. L. **Experiências com desenvolvimento ágil**. São Paulo: IME-USP, 2008. 170 p. Dissertação (Mestrado em Ciências da Computação), Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2008.

BECK, K. **Extreme Programming Explained: Embrace Change**. 1999, 224 p. ISBN: 0-201-61641-6, ed. Addison-Wesley Professional, 1999.

BECK, K, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, e Dave Thomas. 2001. **“Manifesto for agile software development”**. Disponível em: <http://agilemanifesto.org/iso/ptbr/manifesto.html>. Acessado em: 08/06/2014 às 9:24 hs

BOEHM, Barry. W. **A spiral model of software development and enhancement**. SIGSOFT Software Engineering Notes, 1986.

BOEHM, Barry. W. **Spiral Development: Experience, Principles, and Refinements**. Spiral Development Workshop. edited by Wilfred J. Hansen. July 2000. CMU/SEI-2000-SR-008. SEI (The Software Engineering Institute) - Joint Program Office Copyright 2000 by Carnegie Mellon University.

CARVALHO, B. V. e MELLO, C. H. P. REVISÃO, ANÁLISE E CLASSIFICAÇÃO DA LITERATURA SOBRE O MÉTODO DE DESENVOLVIMENTO DE PRODUTOS ÁGIL SCRUM, In: SIMPOI 2009 - XII Simpósio de Administração da Produção, Logística e Operações Internacionais, 2009, São Paulo: FGV/EAESP, 2009. ISSN: 1518-6539. Disponível em: [http://www.simpoi.fgvsp.br/arquivo/2009/artigos/E2009\\_T00109\\_PCN92031.pdf](http://www.simpoi.fgvsp.br/arquivo/2009/artigos/E2009_T00109_PCN92031.pdf) > Acesso em: 14/04/2014 as 14:55 hs

COCKBURN, A. **Agile Software Development: The Cooperative Game**. ed. Addison Wesley Professional, 2nd edition, 504 p., ISBN: 0321482751, 2006.

DAVENPORT, T., “**Putting the Enterprise into the Enterprise System**”, Harvard Business Review, 1998.

DAVENPORT, T., “**Missão Crítica: Obtendo Vantagem Competitiva com os Sistemas de Gestão Empresarial**”, Porto Alegre: ed. Bookman, 2002

DYBA, T. **Improvisation in Small Software Organizations**. Published in Journal IEEE Software [archive](#), Volume 17 Issue 5, Page 82-87, **IEEE Computer Society Press** Los Alamitos, CA, USA, [table of contents](#) doi>[10.1109/52.877872](https://doi.org/10.1109/52.877872), September 2000.

HANSEN, W. J.; FOREMAN, J. T.; CARNEY, D. J.; FORRESTER, E. C.; GRAETTINGER, C. P.; PETERSON, W. C.; PLACE, P. R. **Spiral Development - Building the Culture**. CSE-SEI-WORKSHOP. July 2000. CMU/SEI-2000-SR-006. SEI (The Software Engineering Institute) - Joint Program Office Copyright 2000 by Carnegie Mellon University.

LARMAN, C. **Agile and Iterative Development: A Manager’s Guide**. Addison-WesleyProfessional, 368 p., ISBN: 0131111558, 2003.

LAUDON, K. C. e LAUDON, J. P., “**Sistemas de Informações Gerenciais: Administrando a Empresa Digital**”, 5ª. Ed., ed. Prentice Hall, Brasil, 2004

MELO, C. O. e FERREIRA, G. R. M. Adoção de métodos ágeis em uma instituição pública de grande porte – Um estudo de caso. In: Workshop Brasileiro de Métodos Ágeis (Agile Brasil 2010). Centro de Eventos da PUCRS (CEPUC), Porto Alegre. 14 p. 2010.

MONSORES, M. “**SOFTWARE LIVRE E SISTEMAS ERP: LEVANTAMENTO ANALÍTICO E PROPOSTA DE METODOLOGIA DE PRE-IMPLANTACAO**”, Dissertação (Mestrado em Informática), Departamento de Informática Aplicada, UNIRIO/RJ, Rio de Janeiro, 2009

PRESSMAN, R. S. **Engenharia de software**. 1056p. 3a. ed., ISBN 8534602379. Makron Books. 1995.

PRESSMAN, R. S. **Engenharia de software**. 720p., 6. ed., ISBN 8586804576, ed. McGraw-Hill. 2006.

ROYCE, W. W. Managing the development of large software systems. In Proceedings of **IEEE Wescon**, Aug 1970.

SCHWABER, K. and SUTHERLAND, J. 2013. **The Scrum Guide™ The Definitive Guide to Scrum - The Rules of the Game**. Disponível em <https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf> - acessado em 09/06/2014 as 11:00 hs

SOMMERVILLE, I. **Engenharia de software**. 552 p., 8.ed., ISBN 8588639289, ed. Addison Wesley, 2007.