DESENVOLVIMENTO ÁGIL

Dionatã Dorneles Mafaldo  
Guilherme Legramante Martins  
Jean Trindade Piagetti  
Naihara Sousa Amorim

Raul Vitor Costa Lessa

Universidade Federal do Pampa, Campus Alegrete, RS.

RESUMO

Este Artigo Cientifico tem por finalidade expor uma análise aprofundada sobre o estudo dos métodos de desenvolvimento Ágil de Software. Atualmente a demanda por Software com qualidade é muito grande, mas infelizmente não há profissionais o suficiente para produzir software com qualidade, e a maioria dos profissionais trabalham em empresas com metodologias de desenvolvimento com ênfase em burocracias, e estes métodos muito burocráticos acabam por suas vez, engessando todo o desenvolvimento de Software e o deixando muito rígido, e não flexível a mudanças, sendo que as mudanças são inevitável ao longo da vida útil de um Software, para resolver estes problemas são adotados métodos ágeis de desenvolvimento de Software, cujo estes se apresentam de forma revolucionária, quebrando paradigmas de métodos tradicionais e conservadores. O manifesto Ágil assinado em fevereiro de 2001 por 17 desenvolvedores em Utah, nos traz uma certa reengenharia de Software, visando a agilidade e qualidade do mesmo com grande percentual de assertividade e qualidade do produto final de Software.

**PALAVRAS-CHAVE:** software; métodos; desenvolvimento; ágil; xp; scrum.

**ABSTRACT**

This Article Scientific aims to expose an in-depth analysis of the study of Agile Software development methods. Currently the demand for software quality is very large, but unfortunately there is professional enough to produce software quality, and most professionals working in companies with development methodologies with emphasis on bureaucracy, and these very bureaucratic methods end up your time, plastering whole development of Software and leaving him very hard, not flexible to change that is inevitable over the useful life of the software, to solve these problems are adopted Agile Software Development, whose these are revolutionary , breaking paradigms of traditional methods and conservative. The Agile manifesto signed in February 2001 by 17 developers in Utah, it brings a certain re-engineering software aimed at the same speed and quality with a high percentage of assertiveness of the final software product.

**KEYWORDS:** software; methods; development; agile; xp; scrum

**1. INTRODUÇÃO:**

O que você vera a seguir neste artigo, será uma síntese elaborada através de estudos bibliográficos sobre o tema "Desenvolvimento Ágil", começando por uma introdução de processos de software como pré-requisitos para um pleno entendimento à metodologia ágil de software, o que é, para que serve, os exemplos de metodologias ágeis e as suas principais técnicas utilizadas, a sua aplicação, ferramentas, desafios, benefícios e malefícios.

**1.1 Processos de software**

Para um melhor entendimento de Métodos Ágeis de Desenvolvimento de Software, é importante que se entenda um pouco sobre processos de software, que por sua vez antecedem os métodos ágeis de software.

Um processo de software (ou metodologia de desenvolvimento de software) segundo Sommerville 8ªEd, cap4, pág42: "(...) é um conjunto de atividades que leva à produção de um produto de software." Dentre as várias atividades associadas ao desenvolvimento de software, existem por exemplo o enunciado do problema, o levantamento de requisitos, a análise de requisitos, a viabilidade dos requisitos e a codificação, entre outros... O resultado do processo, é um produto de software que reflete na forma de como a metodologia de desenvolvimento de software foi implementada e conduzida. Segundo Sommerville 8ªEd, cap4, pág42: "Os processos de software são complexos e, como todos os processos intelectuais e criativos, dependem de julgamento humano."

Embora existam vários processos para o desenvolvimento de software, não existe um modelo ideal para cada necessidade de software, sendo que um modelo é apenas uma abstração de um processo de software sob determinada perspectiva e fornece apenas informações parciais sobre o processo, por isto varias empresas e profissionais implementam e utilizam modelos de processos completamente diferentes, e adaptados para as suas necessidades de requisitos e funcionalidades, porém é observado que existem atividades fundamentais comuns a todos eles:

**1. Especificação de Software:** Definição das funcionalidades (requisitos) e das restrições do software. Geralmente é a fase que de preferência o Engenheiro de Software conversa direto com o cliente, entrevistando-o com o intuito de explicitar e tornar palpável o que o cliente deseja, para definir os requisitos e características do novo software.

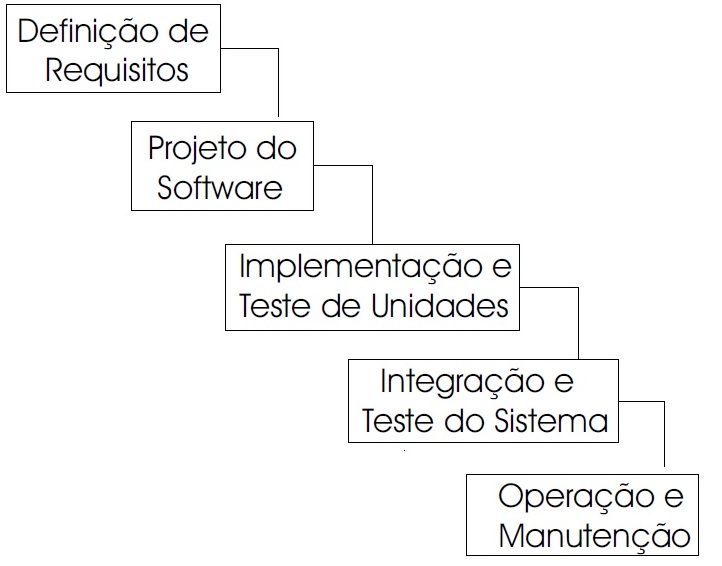
**2. Projeto e Implementação de Software:** O software é produzido de acordo com as especificações. Nesta fase são propostos modelos através de diagramas, definição de viabilidade de requisitos e Implementação.

**3. Validação de Software:** O software deve ser validado para garantir que ele faça o que o cliente deseja. Durante esta fase, serão feitos diversos testes para garantir a qualidade e funcionalidade do software.

**4. Evolução do software:** Para não tornar-se obsoleto o software deve evoluir para atender as necessidades do cliente, necessidades estas que inevitavelmente mudam constantemente. Durante toda a vida útil do software surgirão novas necessidades que deverão ser incorporadas e implementadas ao software e novamente serem testadas para garantir a qualidade do software.

Três modelos de processos tradicionais amplamente usados em engenharia de software, em especial para desenvolvimento de sistemas de grande porte, e que fazem contraposição com os métodos ágeis de software que veremos mais adiante, são:

* **Modelo em cascata** - O primeiro modelo de processo de desenvolvimento de software originou-se devido ao processo de encadeamento de uma fase com a outra, este modelo é conhecido como Modelo em cascata ou ciclo de vida de software.
* **Desenvolvimento evolucionário** - Quando não há a noção explicita dos requisitos, o sistema evolucionário visa atender as necessidades imediatas, e as especificação vão sendo entregue de forma incremental conforme se entenda melhor seus problemas e necessidades.
* **Engenharia de software baseado em componentes** - Este modelo Baseia-se em reuso de software.



**Figura 1:** Estágios doModelo Clássico ou Cascata

**Fonte:** Extraído de Sommerville (8ª Ed.)

**2. METODOLOGIAS ÁGEIS**

O termo “Metodologias Ágeis” tornou-se conhecido em 2001 quando dezessete especialistas em processos de desenvolvimento de software, apresentaram os métodos ágeis para desenvolvimento de software, nos quais, entre eles os que mais se destacaram foi o método Scrum, e Extreme Programming(XP), e também nesta ocasião, estabeleceram princípios comuns compartilhados por todos os métodos ágeis de desenvolvimento de software. Foi então criada a Aliança Ágil e o estabelecimento do “Manifesto Ágil”.

Os conceitos chave do “Manifesto Ágil” são:

**Indivíduos e interações** mais que processos e ferramentas.  
**Software em funcionamento** mais que documentação abrangente.  
**Colaboração com o cliente** mais que negociação de contratos.  
**Responder a mudanças** mais que seguir um plano.

O “Manifesto Ágil” prioriza os conceitos citados acima visando maximização de resultados, mas não rejeita os processos tradicionais de desenvolvimento de software, com seus processos e ênfase em ferramentas, burocracia, documentação, negociação de contratos e seguir planos, mas realmente vê a necessidade de deixá-los em segundo plano, porque identifica que sua importância é secundaria quando comparado com os **indivíduos e interações**, com o **Software em funcionamento**, com a **colaboração do cliente** e as **respostas rápidas a mudanças e alterações**. Porém esses conceitos ágeis, se encaixam melhor as necessidades de pequenas e médias organizações que necessitam de agilidade, qualidade de software e adaptar-se constantemente a mudanças.

\\ "lembrando" < O artigo deve conter as seguintes nove seções:

Resumo

1. Introdução

2. Definição

3. Aplicação

4. Ferramentas

5. Benefícios

6. Desafios

7. Conclusões

Referencias >

**Referências**

# [Sommerville (8° Ed.)] Sommerville,

# [Manifesto para Desenvolvimento Ágil de Software] Disponível em http://agilemanifesto.org/iso/ptbr/, acessado em 18 de Junho de 2015.