

# Boas Práticas na Gestão de Mudança de Metodologia Cascata para Metodologia Ágil em Grandes Empresas

Esther Baldim Guerra

**Orientador:** Prof. Dr. Luciano Silva

**Resumo.** Este artigo tem por objetivo abordar e explorar como a transição de metodologias de projeto pode ser realizada da melhor maneira tanto para organização quanto seu colaborador. Inicialmente serão estudados dois dos modelos de processo existentes, sendo eles o Modelo de Processo Prescritivo e o Modelo de Processo incremental, atentando somente para uma metodologia de cada modelo, sendo elas respectivamente a Metodologias Cascata e Ágil, aprofundando no método Scrum para organização de equipes, gerenciamento das entregas, escopo e requisitos do projeto.

**Abstract.** This article aims to address and explore how the transition of project methodologies can be carried out in the best way for both the organization and its collaborator. Initially, two of the existing process models will be studied, namely the Prescriptive Process Model and the Incremental Process Model, paying attention only to one methodology of each model, namely the Cascade and Agile Methodologies, deepening the Scrum method for team organization. management of deliveries, scope, and requirements of the project.

## 1. Introdução

Para interpretar as existentes metodologias de projeto de software é necessário primeiro entender o que é a Engenharia de Software e o que se almeja em uma entrega e desenvolvimento de um software, para que assim seja possível tornar tangível o quanto uma determinada metodologia atende as necessidades do cliente tanto quanto do seu desenvolvedor, que espera entregar algo funcional, de sustentação prática e principalmente atendendo as solicitações recebidas. A Engenharia de Software é uma área de estudo que se atenta à todas as partes envolvidas na produção de software, desde seu estado inicial até a documentação final de todo processo, para tanto é necessário seguir um modelo de processo.

Segundo Sommerville (2011) o software é o conjunto do programa mais a sua documentação, esse programa por sua vez pode fazer parte de um sistema, logo uma documentação bem estruturada é necessária. Para se desenvolver um Software de valor a engenharia do mesmo precisa focar em certos requisitos como confiabilidade, segurança e resiliência, mas principalmente atender os pedidos realizados pelo cliente, num nível aceitável de performance.

Os modelos de processos segundo Pressman (2016) surgiram para organizar o caos que havia na área de desenvolvimento de software, eles são um guia detalhado do fluxo de todas as atividades, ações e tarefas, o grau de iteração, os artefatos e a organização do trabalho a ser feito. Será feita a comparação entre duas metodologias ao longo desse trabalho, a primeira de um Modelo de Processo Prescritivo, ou seja, onde as atividades ocorrem de maneira sequenciada, e a segunda de um Modelo de Processo Incremental onde se realiza pequenas entregas de valor atendendo primeiramente os requisitos básicos do produto.

Este trabalho aborda o problema de transição do processo tradicional de desenvolvimento de software (Cascata) para um processo ágil (SCRUM), evidenciando seus principais problemas e propondo um conjunto de boas práticas para abordar tais problemas. Infelizmente, na literatura de Engenharia de Software, não existe um conjunto sistematizado de boas práticas para realizar esta transição.

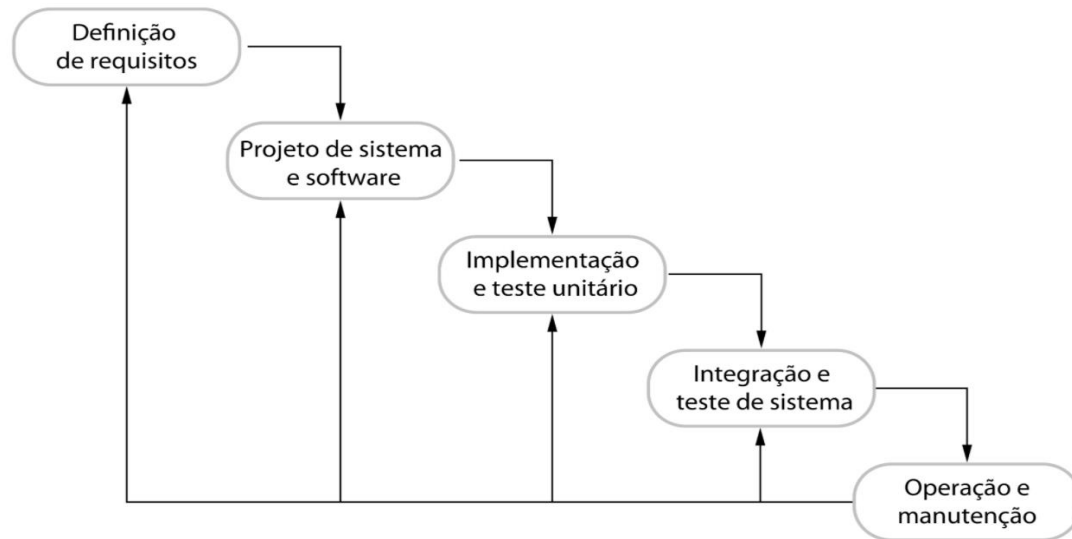
O artigo está organizado da seguinte forma:

- a Seção 2 trata do processo ou metodologia de desenvolvimento de software cascata;
- a Seção 3 detalha o processo SCRUM;
- a Seção 4 expõe o problema de transição destas metodologias;
- A Seção 5 mostra um conjunto de boas práticas para esta transição
- E finalmente, a Seção 6 apresenta as conclusões do trabalho.

## **2. Metodologia Cascata**

A Metodologia Cascata é baseada em especificar todas as etapas anterior ao desenvolvimento do software, ou seja, todos integrantes do projeto devem entender com clareza as atividades, os requisitos e objetivo para ter um desenvolvimento adequado, um bom mapeamento de requisitos e testes, igualmente a entrega do produto.

Na figura abaixo pode-se observar as fases que constituem a Metodologia Cascata, resalta-se que existem variações dos nomes de cada etapa de acordo com a empresa ou parte que está fazendo uso da metodologia, porém, o princípio de que uma fase não pode ser iniciada antes que outra termine e que todas devem ser mapeadas no início do projeto, não se altera.



**Figura 1** – Etapas do Modelo em Cascata – 9ª Edição, Engenharia de Software  
*Fonte: (Sommerville, 2011)*

1. *Definição de requisitos.* Em contato com o usuário se definem as limitações, metas e serviços que serão necessários no sistema, esse detalhamento servirá como documentação.
2. *Projeto de Sistema e Software.* Momento em que a arquitetura geral do sistema permite a destinação de recursos, segundo Sommerville, o projeto de software envolve mapeamento e descrição de forma conceitual do sistema de software e suas interligações.
3. *Implementação e teste unitário.* O teste unitário é projetado para averiguar se cada parte desenvolvida funciona corretamente.
4. *Integração e teste do sistema.* As partes do sistema são integradas e testadas, observando atentamente se os requisitos foram atendidos, tendo um resultado eficaz, entrega-se o produto ao cliente.

5. *Operação e manutenção.* Considerada por muitos a maior fase do projeto. Nessa fase o sistema é inserido para uso, o que resulta no encontro de erros e falhas que não foram mapeadas no princípio do projeto e precisam de correção.

Nesse cenário não é praticável ter uma boa resposta às eventualidades, situações não previstas do dia a dia dos participantes do projeto e da própria empresa, além de não ser factível o cliente ter completo conhecimento de suas necessidades uma vez que elas estão em constante crescimento e reestruturação para atender o mercado, sendo assim qualquer mudança de escopo, adição de novos requisitos, melhoria ou erro de processo faz com que seja necessário replanejar por completos as atividades, o escopo, e toda a base no que foi construído o projeto, resultando em atrasos, além do que, por não ter uma comunicação aberta com o cliente durante o desdobramento do projeto a possibilidade de entregar algo que não possui valor, cresce.

### 3. Metodologia Ágil focada em Scrum

Hoje é necessário que as entregas sejam realizadas na mesma velocidade em que são recebidas via pedido do cliente. As empresas estão passando por readaptação constante para conseguir se manter no mercado. A metodologia ágil tem o entendimento de que não é viável mapear o projeto como um todo antes de se iniciar o desenvolvimento, isso não significa que não há planejamento do escopo e requisitos, mas sim que o maior objetivo é entregar algo de valor para o cliente, sendo com requisitos básicos ou não. Na figura abaixo tem-se os pilares da metodologia.

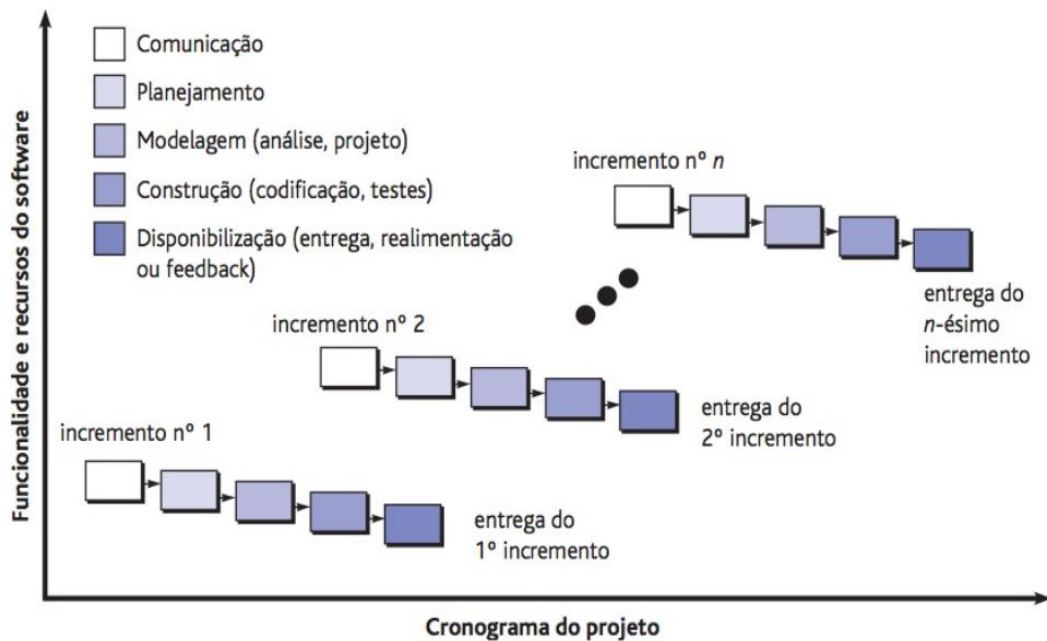
Princípio	Descrição
Envolvimento do cliente	Os clientes devem ser envolvidos em todo o processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos de sistema e avaliar as iterações do sistema.
Acolher as mudanças	Tenha em mente que os requisitos do sistema mudam e, portanto, deve-se projetar o sistema para acomodar essas mudanças.
Entrega incremental	O software é desenvolvido em incrementos, e o cliente especifica os requisitos incluídos em cada um deles.
Manter a simplicidade	Deve-se ter como foco a simplicidade, tanto do software que está sendo desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.
Pessoas, não processos	As habilidades do time de desenvolvimento devem ser reconhecidas e aproveitadas da melhor maneira possível. Seus membros devem ter liberdade para desenvolver seu modo próprio de trabalhar sem se prender a processos determinados.

**Figura 2** – Pilares estruturais do Modelo Ágil – 9ª Edição, Engenharia de Software

Fonte:(Sommerville, 2011)

Um dos principais pontos dessa metodologia é o acompanhamento do cliente em todas as partes do projeto, seja na especificação dos incrementos tanto quanto na avaliação deles, ademais é possível realizar a solicitação de implementação de novos requisitos, que serão compreendidos e enxertados no planejamento conforme entendimento do Gerente do Projeto.

As fases da Metodologia Ágil são especificação, desenvolvimento e implementação ocorrendo paralelamente, observe a figura a seguir para melhor compreensão e visualização desse processo.



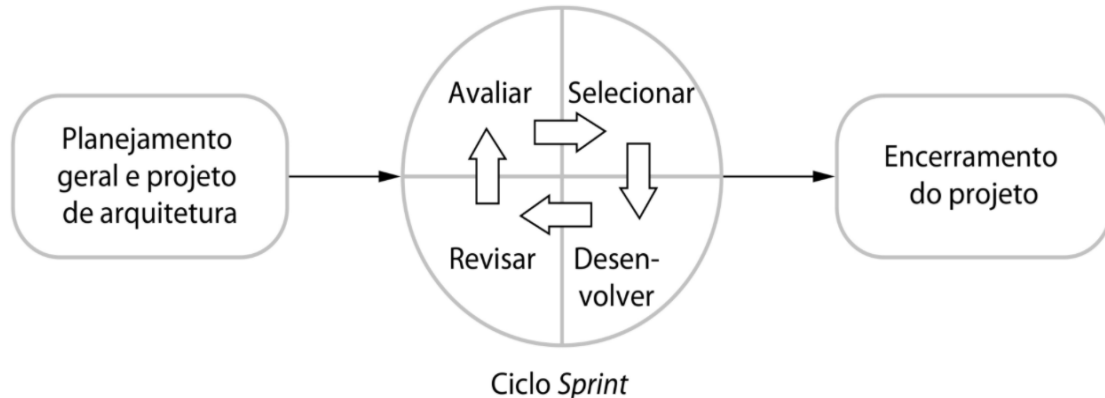
**Figura 3** – O Modelo incremental – 2016, Engenharia de Software: uma abordagem profissional

Fonte: (Pressman, 2016)

Uma visão cascata traz uma certeza para os gerentes de grandes empresas do que está sendo desenvolvido, quem está atuando no processo e quando será entregue, contrariamente no ágil não se possui essa visibilidade pois as equipes são autogerenciáveis e as entregas são pequenas e estipuladas pela própria equipe, assim surgiu o framework Scrum, para que as empresas que não faziam uso originalmente da metodologia ágil possam se adaptar a mesma, sabendo o que está sendo entregue, quanto estão entregando e se os prazos serão atendidos.

De acordo com Sommerville o método Scrum pode ser entendido em três fases sendo a primeira um planejamento geral onde se entendem os objetivos e arquitetura a ser desenvolvida, seguido pelo ciclo de *sprints* considerando uma

entrega em cada um desses ciclos individualmente, encerrando então com a entrega do projeto tendo uma documentação clara e objetiva para apoiar o cliente além de manuais para o usuário final e uma avaliação do aprendizado por parte da *squad*.



**Figura 4** – O processo Scrum - 9ª Edição, Engenharia de Software

Fonte: (Sommerville, 2011)

O Scrum segundo Sommerville (2011) não é um método convencional para se gerenciar projetos, então foi-se estabelecida uma terminologia, como por exemplo, *Scrum Master* para substituir a figura 'Gerente de Projetos'. Por não ter o emprego de práticas de desenvolvimento específicas ele pode ser utilizado e adaptado naturalmente as empresas que estão em transição de metodologias e necessitam realizar a gestão dessa mudança.

As etapas ou como são chamadas, cerimônias do Scrum podem ser entendidas da seguinte maneira:

- *Sprint*: tempo determinado para se acontecer pequenas entregas, podem variar de uma a duas semanas e ao final desse período deve-se avaliar o que foi desenvolvido e o valor que foi acrescentado ao produto.
- *Retrospective*: é quando a própria equipe se avalia quanto a desenvoltura nas atividades, aqui também são avaliados os problemas que surgiram na *Sprint* e qual a melhor forma de endereçá-los, dessa conversa precisa-se sair com um plano de ação para os pontos de atenção identificados.
- *Review*: entende-se como um momento de avaliação do desenvolvimento do código ou documentação e apresentação da entrega aos demais envolvidos no projeto.

- *Daily*: tem a duração de 15 minutos, geralmente acontece no início do dia, porém a equipe pode escolher qual o melhor horário e é necessário que cada participante responda as seguintes questões de forma mais breve possível: “O que você fez ontem?”, “Encontrou algum problema?” e “O que vai fazer hoje?”
- *Planning*: realizada no início de cada Sprint, é o momento em que o dono do produto (conhecido como PO – *Product Owner*) prioriza as atividades para aquela Sprint apenas, entendendo qual delas irá agregar maior valor a entrega.

A equipe no Scrum recebe a terminologia em inglês *Squad*, os papéis de quem faz parte dela são *Product Owner*, este é quem especifica quais atividades são mais relevantes e também tem a especificação de quais requisitos a entrega deve atender para ser aceita, o *Scrum Master* ou *Team Lead*, tem a responsabilidade de direcionar a equipe quanto a em que etapa o projeto se encontra, dar o direcionamento nas cerimônias do *Scrum*, além de prestar apoio ao PO, fornece indicadores de previsão da entrega e remove empecilhos, a Equipe de Desenvolvimento por sua vez é responsável pela realização das atividades, entendimento dos requisitos, testes e entregas.

#### **4. O Problema da Mudança de Metodologias**

Observa-se que há diferentes formas para classificação de porte de empresas, nesse trabalho está sendo considerado a capacidade econômica segundo o Banco Nacional do Desenvolvimento (BNDES) para determinar o porte da organização.

O que se encontra no mercado são Grandes Empresas fundadas em Metodologia Cascata buscando uma rápida transição para a Metodologia Ágil por entender que os concorrentes que a seguem tem maiores e mais valorosas entregas. Vale ressaltar que as metodologias de projetos não dizem respeito somente a como serão desenvolvidas as atividades do projeto, como serão quantificados os indicadores ou ainda de que forma será realizada a documentação, mas também como se deve estruturar o papel de cada colaborador dentro da organização, observando como o Scrum descreve as equipes será entendido qual o problema em querer mudar de forma repentina de uma estrutura cascata para ágil.

No Scrum as equipes devem ser auto-organizáveis, autogerenciáveis e multifuncionais, ou seja, a *Squad* determina o quanto consegue entregar, em

quanto tempo, como desenvolver cada um dos requisitos e estruturar a arquitetura do Software, além do mais dentro dessa Squad deve-se ter todas as partes essenciais para se completar o projeto com eficiência.

Segundo relato próprio, no momento de transição o desenvolvedor final recebe a informação de um papel superior em sua organização de que sua forma de trabalho vai mudar drasticamente e nas diversas vezes essa notícia vem como uma surpresa um tanto desagradável, afinal ele deverá seguir uma nova metodologia que este pode ou não ter conhecimento sobre e essa situação afeta diretamente suas entregas, causando desconforto e resistência de sua parte.

Ao passo que aquele que definiu a nova forma de trabalho também recebeu essa notícia de um cargo superior e pode não estar preparado para a nova metodologia, no momento ele está habituado a ter uma forma de controle sobre as entregas do projeto, alocação de recurso, reconhecimento de riscos, avaliação de resultado e desenvolvimento das atividades, com uma mudança de metodologia deve existir uma transformação na forma de pensar, seja sobre como se organiza a equipe ou como se extrai informações sobre o projeto.

O entendimento sobre a metodologia tanto de forma técnica como também sobre de que forma aplicar na vida de trabalho diária traz segurança para aquele que aplica, porém não é dessa forma que está sendo feita a mudança. Por falta de conhecimento, se torna difícil acreditar na nova metodologia e apoiar sua implementação, mesmo seguindo o método Scrum onde os participantes do projeto ganham autonomia.

Mesmo com as inseguranças dos trabalhadores a metodologia é aplicada forçosamente o que acarreta debilidades em pontos chaves da Metodologia Ágil, como por exemplo, a comunicação com cliente não ser clara e objetiva.

## **5. Boas Práticas na Mudança de Metodologias**

A seguir serão expostas fases reconhecidas a partir de experimento pessoal que apoiam a gestão da transição da Metodologia Cascata para Metodologia Ágil.

- *Capacitação.* A organização fornece ao colaborador treinamento técnico e prático sobre a nova metodologia, com entrega de certificado válido no mercado, assim aumentando a confiança do mesmo em implementar o modelo de processo do qual terá total domínio técnico, essa capacitação também deverá acontecer de forma tempestiva, ou seja, com um período de minimamente duas semanas antes de este precisar colocá-la em prática.
- *Aplicação.* Tarefas práticas em um projeto real para que o colaborador tenha sua primeira experiência seja como líder, desenvolvedor ou



participante com conhecimento do processo sobre o qual está sendo realizado o projeto.

- *Entendimento de resultados.* Ao final da experiência, todos participantes do projeto devem ser capazes de apresentar os pontos fortes e debilidades da equipe na implementação da metodologia.
- *Apresentação de indicadores.* Assim apresenta-se quantitativamente os ganhos, sejam eles financeiros, de tempo de trabalho, melhoria da imagem da empresa, aumento da segurança do processo em que o projeto é focado, ou qualquer outro ganho. Importante lembrar que a forma como os indicadores serão apresentados é de responsabilidade da equipe.

## **6. Conclusões**

Enfim é esperado que com o entendimento e experiências providos o colaborador possa ter plena capacidade de desempenhar suas funções diárias sem grandes empecilhos. Durante todas as fases é fundamental que os líderes de capacitação sejam incentivadores e principalmente acreditem eles mesmos na capacidade de cada colaborador em desenvolver essa nova habilidade em projetos, além de entender as limitações de cada participante e prover suporte para as atividades de forma didática e sucinta.

A transição sendo iniciada de forma clara e objetiva diminui ou até mesmo pode mitigar as inseguranças em relação a mesma, ressalto que a continuidade em apoio e capacitação são de extremamente necessidade para se obter um bom desenvolvimento pessoal do colaborador.

Entende-se que conhecimento técnico é algo necessário, mas o conhecimento adquirido com prática se torna firme e confiável logo sendo possível passar a diante. A cultura empresarial precisa andar de mãos dadas com a nova mudança, não é possível implementar algo novo e não querer mudar. A transição de metodologias é desconfortável, mas entendido que se há mudança e transformação também não há crescimento e expansão, ou seja, antes de realizar algo tão renovador é necessário avaliar o custo, a aceitação das partes, mas principalmente saber adaptar, as metodologias ágeis surgiram para apoiar, facilitar e melhorar a vida do empregador e do empregado.

## Referências Bibliográficas

Beck, Kent; Beedle, Mike; Bennekum, Arie Van; Cockburn, Alistair; Cunningham, Ward; Fowler, Martin; Grenning, James; Highsmith, Jim; Hunt, Andrew; Jeffries, Ron; Kern, Jon; Marick, Brian; Martin, Robert C.; Mellor, Steve; Schwaber, Ken; Sutherland, Jeff; Thomas, Dave; **Manifesto for Agile Software Development** – 2001.

BNDES – O Banco Nacional do Desenvolvimento – Disponível em: <https://www.bndes.gov.br/wps/portal/site/home/financiamento/guia/porte-de-empresa>. Acesso em 20.04.2021.

Pressman, R. S.; Maxim, B. R. **Engenharia de Software: uma Abordagem Profissional** – 8 ed., Porto Alegre: AMGH, 2016.

Royce, W.W. **Managing the Development of Large Software Systems**. Disponível em: <http://www-scf.usc.edu/~csci201/lectures/Lecture11/royce1970.pdf>. Acesso em 01.03.2021.

Sommerville, I. **Engenharia de Software** – 9 ed., São Paulo: Pearson Prentice Hall, 2011.

Sutherland, J. **Scrum: a arte de fazer o dobro do trabalho na metade do tempo**; tradução Nina Lua. – 2 ed. – São Paulo: Leya, 2016.