DESENVOLVIMENTO ÁGIL DE SOFTWARE

Trabalho Avaliativo - Etapa 01

Desenvolvimento de uma Metodologia Híbrida

Professor: Diego Garcia

Alunos:

Ícaro Bicalho Quintão EC - 14.1.8083

Pedro Henrique Mendes Batista EC - 14.1.8403

Bruno Passamai Alvarenga EC - 13.2.8458

Pedro Henrique Ferreira Loures EC - 16.1.8203

Hugo Carvalho Dias Ferreira SI - 12.2.8014

Lucas Pereira Miranda EC - 17.2.8207

Innan Plínio Rangel Amorim EC - 16.2.8416

Edgar Henrique Alves Rodrigues EC - 15.1.8029

Leilisson Figueiredo EC - 15.2.5812

Marlon da Silva Martins SI - 16.2.8361

Gabriella Mendonça Santa Clara EC - 14.1.8200

Gabriel Oliveira Silva EC - 15.1.8292

Thiago Duarte Pinto Coelho SI - 16.1.8132

Rosabel Vieira Braga SI - 17.2.5909

João Monlevade

Agosto 2019

# Sumário

[**Sumário**](#_lkvaqd5q711b) **2**

[**Introdução**](#_7znxzganjw66) **3**

[**Metodologia Clássica**](#_ic2nrscgkt6m) **3**

[**Metodologia Ágil**](#_dtd2cqm6xcx0) **4**

[**Metodologia Clássica versus Ágil**](#_p4gnsn9jmpjd) **5**

[**Metodologia Desenvolvida**](#_ayn7dcmmvp1s) **7**

[**Conclusão**](#_ywqnufig09px) **8**

[**Referências**](#_wpnrlft0pev8) **9**

# Introdução

Podemos definir um projeto de software como sendo um empreendimento temporário dividido em ciclos de vida ou fases com o objetivo de criar um produto, serviço ou resultado único de forma a atender às especificações e expectativas de negócio. Os ciclos de vida de um projeto de software normalmente definem: qual trabalho técnico será realizado, quais entregas serão geradas em cada fase, como serão verificadas e validadas, quais as pessoas ou papéis estarão envolvidos em cada fase e como será realizado o controle e aprovação de cada fase.

O desenvolvimento de software é uma tarefa muito complexa, difícil e de alto risco. Inúmeros problemas são enfrentados diariamente. Entre alguns desses problemas enfrentados, podemos destacar: alto consumo de tempo que supera estimativas e cronograma, funcionalidades que não solucionam os problemas dos usuários, baixa qualidade dos sistemas desenvolvidos, tempo para teste e qualidade reduzidos e, em alguns casos, até o cancelamento do projeto por inviabilidade.

Tendo como grande objetivo a redução de risco associado ao desenvolvimento de software, as metodologias definem uma maneira de trabalhar em projetos utilizando processos, sendo elas um conjunto de atividades e resultados associados que auxiliam no desenvolvimento e construção de softwares.

# Metodologia Clássica

O modelo clássico ou cascata (waterfall), que também é conhecido como abordagem “*top-down*”, foi proposto por Royce em 1970, derivado de modelos industriais de diversos segmentos de engenharias. Seu principal objetivo era de estabelecer ordem e padrão em desenvolvimento de software. Foi chamado de desenvolvimento tradicional, pois é a base para diversos modelos utilizados há décadas pela indústria de software e é considerado um modelo rígido de pouca flexibilidade, adaptabilidade e versatilidade, sendo utilizado em projetos de pequeno, médio e grande porte.

Grande parte de seu alto grau de aceitação e utilização está no fato de ser um modelo orientado para documentação, seja ela de qualquer tipo, como textos, representações gráficas, simulação, diagramas, casos de usos, entre outras. A ideia principal dessa metodologia é que o software é construído baseado em uma sequência de fases, sendo que cada uma delas depende da conclusão da outra para ser iniciada, com exceção da primeira, conforme ilustra a Figura 1.



Figura 1. Ciclo de desenvolvimento utilizando metodologia tradicional

Os resultados da primeira etapa, da Análise, são concluídos, então a saída “flui” para a segunda etapa, que é a de Projeto. Logo, quando a etapa de Projeto for concluída, sua saída ou resultado “flui” para a seguinte, sendo que uma etapa nunca se inicia antes da etapa anterior ser finalizada. As atividades são agrupadas em tarefas executadas sequencialmente.

# Metodologia Ágil

A metodologia ágil surgiu da necessidade de minimizar riscos e custos associados ao desenvolvimento de software. Diferentemente da metodologia clássica, a atividade de testes, por exemplo, no sistema ágil é um processo empírico, sendo realizado em todas as fases do projeto, do início ao fim, onde os profissionais de testes contribuem para o processo como um todo, validando os requisitos desde a sua criação até a entrega final. Como não há necessidade explícita de uma documentação extensa e abrangente, a comunicação entre a equipe é fator crucial de sucesso. Existe a validação de documentos em todos os passos e momentos do processo e isso contribui para o aprendizado de todos, já que todos participam de todo o processo.

# Metodologia Clássica versus Ágil

A metodologia tradicional é fundamentada em processos definidos e a metodologia ágil em processos empíricos. De forma mais resumida e direta, podemos observar na tabela abaixo, e também na Figura 2, o que foi discutido até aqui, como por exemplo a distribuição das etapas em cada método e enfatizando as diferenças e características entre os dois tipos de metodologias, sendo que todos os pontos levantados afetam direta ou indiretamente o processo de testes e a forma como e quando são executados.

Metodologias tradicionais costumam ser mais indicadas em cenários mais formais, além de serem mais recomendadas caso a equipe seja grande e os sistemas a serem desenvolvidos sejam de alto risco (muitas vezes requerem mais documentação). Já os métodos ágeis são normalmente mais recomendados para equipes pequenas nas quais a flexibilidade diante de constantes mudanças é fundamental.

|  |  |
| --- | --- |
| METODOLOGIA CLÁSSICA | METODOLOGIA ÁGIL |
| 1. Processos definidos 2. Planejamento rígido 3. Maior foco em processos do que no produto 4. Feedbacks não são essenciais 5. Documentação extensa 6. Inibe a comunicação entre as pessoas 7. Planejamento prevê um trabalho extenso, com a entrega do produto somente nos estágios finais do cronograma (não evita conflitos com cliente) 8. Conceito de que “entradas iguais sempre geram saídas iguais” 9. Atividades realizadas sequencialmente. Testes executados somente no final, quando “tudo” estiver pronto 10. Baseado na definição de todas as etapas do trabalho 11. Resistência a mudanças 12. Decisões tomadas em uma abordagem top-down 13. Forte centralização em torno da figura do gerente de projetos 14. Liderança que monopoliza toda a comunicação já que a preocupação é com o controle das ações 15. Líderes indicando “O que fazer” e “Como fazer”, ao invés de dizer o “Porquê” 16. Problemas geralmente escalados até a gerência 17. Longa fase de análise; em muitos casos parte da equipe é deixada de lado nesses estágios iniciais (já que considera que tais membros ingressaram apenas na fase de execução) 18. Um forte enfoque na geração de documentos e no controle através desses artefatos 19. Maior envolvimento do cliente em estágios iniciais, com certo relaxamento de postura, uma vez que o projeto tenha se iniciado 20. Foco na “antecipação” (algo difícil em um ambiente sempre sujeito a mudanças repentinas) | 1. Processos empíricos 2. Maior liberdade no planejamento das ações 3. Menos formalidade e maior ênfase no produto 4. Feedbacks são essenciais 5. Documentação necessária 6. Estimula a comunicação entre as pessoas 7. Entregas de partes do projeto de forma contínua e incremental (iterações) com o objetivo de obter um rápido feedback do cliente sobre o andamento do projeto 8. Conceito de que “entradas iguais geram saídas diferentes” 9. Atividades realizadas paralelamente. Testes executados durante todo o processo, equipe de testes mais presente em todo os pontos de desenvolvimento 10. Baseado na experiência e controlado através de inspeção e adaptação contínua 11. Flexibilidade e postura positiva diante da necessidade de mudanças (mesmo em fases finais do projeto) 12. Liberdade para o time tomar decisões em conjunto 13. Responsabilidade compartilhada entre os membros da equipe, espírito de colaboração e time engajado 14. Comunicação fluida e livre entre os membros do time 15. Equipes auto organizáveis; a divisão do trabalho é resultado do entendimento do projeto e de um consenso entre o time 16. Atuação conjunta do time para a resolução de problemas 17. Reuniões diárias entre o time onde são discutidos o que será feito naquele momento, revendo o planejamento a médio e curto prazo, além de prováveis impedimentos 18. Embora existam documentos e se estimule a criação dos mesmos, há um pragmatismo maior (sem conferir uma importância exagerada a esses artefatos) 19. Participação ativa do cliente, inclusive enquanto o projeto está sendo implementado 20. Ênfase na “adaptação” (requer “jogo de cintura”) |

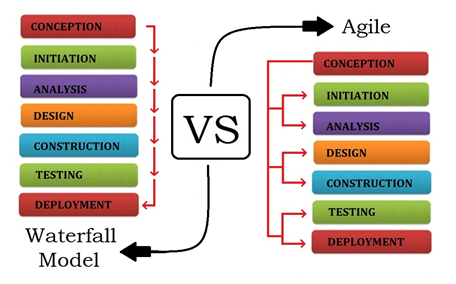


Figura 2. Metodologia clássica x metodologia ágil

# Metodologia Desenvolvida

A proposta é combinar princípios tradicionais para prover atividades de planejamento, controle e coordenação, críticos para o desenvolvimento distribuído, e utilizar os princípios ágeis como veículo de resposta para o dinamismo e incertezas dos requisitos do ambiente.

Nossa metodologia é constituída por cinco componentes e um procedimento iterativo contendo sete etapas para o planejamento e o acompanhamento do projeto, sendo os componentes:

* Modelo de Fases e Entregas: um modelo de processo de negócio simplificado, partindo-se da premissa da existência de um conjunto de fases, atividades e artefatos interagindo e realizando o processo de desenvolvimento de produtos;
* Painel Visual de Planejamento e Controle de Projetos: é um painel físico que contém todas as informações referentes ao planejamento e controle das entregas do projeto;
* Quadro de Planejamento Fino Semanal: trata-se de um quadro de planejamento de atividades e pacotes de trabalho num curto período de tempo;
* Sistema de Gestão de Projetos – diferentemente de outras abordagens, o Gerenciamento Ágil de Projetos explora o uso das ferramentas de tecnologia da informação como fontes de dados relacionados a uso dos recursos em ambiente multi-projetos e decisões GO / NO-GO para projetos futuros;
* Sistema de Indicadores de Desempenho: trabalha com um software de gerenciamento de projetos como base de dados, permitindo a construção de medidores e indicadores de desempenho para o acompanhamento e controle dos projetos.

As etapas para aplicação do modelo são:

1. Definir requisitos do produto que se deseja desenvolver, o que consiste usualmente nos serviços que se devem fornecer, limitações, objetivos e estabelecer datas de entrega.
2. Inserir as entregas no PVPCP, utilizando cartões;
3. Inserir dados das entregas no SGP;
4. Decompor as entregas do PVPCP em pacotes de trabalho e atividades;
5. Executar os pacotes de trabalho e atividades e atualizar o SGP;
6. Gerar relatórios de desempenho do projeto utilizando o SGP;
7. Analisar resultados da iteração, verificar aprendizado e progresso do projeto.

# Conclusão

Metodologias possuem seus pontos altos e baixos, as diferenças entre elas são fundamentais para a escolha de uma que irá participar da concepção de um projeto de software. Em síntese, todos os modelos assumem que o projeto irá passar por cada etapa apenas uma vez. A diferença é que o ágil preza por uma melhor comunicação, ciclos incrementais, pouca documentação e retrabalho antes da entrega ao cliente e não após a entrega como funciona no modelo tradicional.

A elaboração de um modelo específico para um determinado projeto dependerá de diferentes variáveis, como por exemplo, as características de cada projeto, o perfil da empresa e do que é requisitado pelo cliente. A elaboração também deve levar em conta a se é possível implantá-la, pois se caso o development team não for familiarizado com a parte relativamente nova de desenvolvimento ágil, pode aumentar drasticamente os gastos, já que a equipe precisará de treinamento, absorção de conhecimento, além da curva de aprendizado. Ambas as abordagens são excelentes dentro do que propõem, mas não perfeitas, e possuem pontos altos e baixos em áreas específicas ou em determinados tipos de aplicações utilizadas em projetos distintos.

Definir uma metodologia ou saber como utilizá-la tem fator de importância decisivo entre obter ou não sucesso. Uma má definição ou, ainda, uma má utilização dos recursos providos por elas aumenta consideravelmente os riscos do projeto não obter êxito. Saber utilizar é tão importante quanto a escolha de qual delas utilizar. A definição da metodologia impacta drasticamente na qualidade, já que a equipe de testes é afetada diretamente. Além disso, a escolha da metodologia também afeta a escolha do perfil de profissionais necessários para compor a equipe.

Com os métodos escolhidos teoricamente podemos ganhar agilidade, tempo e melhoramos a nossa qualidade nas entregas, mas não garantimos obtenção de sucesso.

# Referências

* Processo de teste ágil x Tradicional, disponível em: <<https://www.devmedia.com.br/processo-de-teste-agil-x-tradicional/36854>> Acesso em: Setembro de 2019.
* Introdução ao FDD - Feature Driven Development, disponível em: <[https://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/2797](https://www.devmedia.com.br/introducao-ao-fdd-feature-driven-development/27971)1> Acesso em: Setembro de 2019.