

Pós-Graduação

Projetos ágeis e análise de sistemas

Tema 03 – Métodos ágeis

Bloco 1

Juliana Schiavetto Dauricio



Objetivos

1. Apresentar as metodologias de desenvolvimento ágil mais comumente utilizadas no mercado de análise e desenvolvimento de sistemas.
2. Fornecer subsídios para a identificação, definição e aplicação de uma metodologia de desenvolvimento ágil em análise e desenvolvimento de sistemas.
3. Permitir que sejam conhecidas as boas práticas em análise de sistemas e gestão de equipes de projetos de desenvolvimento de *software* que adotam metodologias ágeis.

Introdução

- Por que projetos com métodos ágeis?
 - **Para acompanhar o dinamismo das empresas em um mercado global.**

O investimento em *softwares* passou a ser uma medida que requer respostas rápidas em termos de análise, desenvolvimento e implantação de sistemas de informação.

Introdução

- Por que projetos com métodos ágeis?
 - **Devido à necessidade de uso de um *software* para apoio a uma determinada tarefa.**

Os processos de especificação, projeto e implementação são intercalados. Não há especificação detalhada do sistema, e a documentação do projeto é minimizada ou gerada automaticamente pelo ambiente de programação usado para implementar o sistema. O documento de requisitos do usuário apenas define as características mais importantes (SOMMERVILLE, 2011, p. 39).



Como é o desenvolvimento ágil na prática?

- **Mais acelerado e busca:**
 - **Realizar os incrementos das funcionalidades do sistema e disponibilizar sua nova versão.**

Tempo investido e sugerido pela engenharia de *software* no planejamento e análise do sistema está em desuso?

- **O planejamento, de acordo com métodos tradicionais de análise e desenvolvimento de sistemas, requer maior tempo no planejamento do sistema.**
- **No entanto, com as demandas em soluções de *software* cada vez mais rápidas e crescentes, podem não atender mais.**
- **Em função desse novo ritmo do mercado, é que a adoção de uma metodologia de desenvolvimento ágil é uma boa opção.**

Solução em sistema em tempo recorde

- 1. É necessário transpor barreiras e estabelecer as prioridades na identificação e especificação de requisitos, bem como, no próprio desenvolvimento de sistemas.**
- 2. Premissas do manifesto ágil incluem a dedicação ao projeto focado em pessoas e as interações que precisam do sistema.**
- 3. É mais valioso para a empresa o *software* em funcionamento do que a documentação abrangente.**

E o cliente?

Exige uma participação **maior** do **cliente** durante **todo o processo** de desenvolvimento, de forma que o **objetivo** esteja mais voltado a **atender às demandas** de acordo com as mudanças ocorridas, **do que** necessariamente em **planos** (SOMMERVILLE, 2011).

Métodos ágeis

- Os princípios dos métodos ágeis:

Princípios	Descrição
Envolvimento do cliente	Os clientes devem estar intimamente envolvidos no processo de desenvolvimento. Seu papel é fornecer e priorizar novos requisitos do sistema e avaliar suas iterações.
Entrega incremental	O <i>software</i> é desenvolvido em incrementos com o cliente, especificando os requisitos para serem incluídos em cada um.

Métodos ágeis

- Os princípios dos métodos ágeis:


Princípios	Descrição
Pessoas, não processos	As habilidades da equipe de desenvolvimento devem ser reconhecidas e exploradas. Membros da equipe devem desenvolver suas próprias maneiras de trabalhar, sem processos prescritivos.
Aceitar mudanças	Deve-se ter em mente que os requisitos do sistema vão mudar. Projete o sistema de maneira a acomodar essas mudanças.

Métodos ágeis

- Os princípios dos métodos ágeis:

Princípios	Descrição
Manter a simplicidade	Focalize a simplicidade, tanto do <i>software</i> a ser desenvolvido quanto do processo de desenvolvimento. Sempre que possível, trabalhe ativamente para eliminar a complexidade do sistema.

Fonte: Sommerville (2011, p. 40).



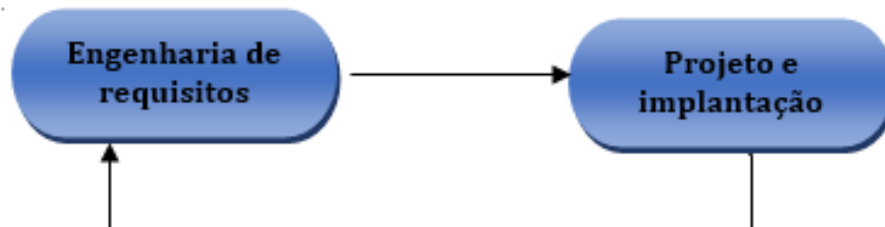
Diferença estrutural no que tange à análise e desenvolvimento de sistemas orientada a planos do desenvolvimento ágil?

Tradicional x Ágil



Figura 1: Especificação dirigida a planos

Fonte: Adaptado de Sommerville, 2011, p. 43.





Leia no material:

Ficam as recomendações de leitura do seu material didático.

Bons estudos!

Pós-Graduação

Projetos ágeis e análise de sistemas

Tema 03 – Métodos ágeis

Bloco 2

Juliana Schiavetto Dauricio





Métodos ágeis

XP e Scrum

Extreme Programming

- Alguns conceitos fortemente envolvidos nesta metodologia de desenvolvimento ágil são:
 - Comunicação.
 - *Feedback*.
 - Desenvolvimento por parte da equipe de trabalho de habilidades relacionadas a comportamentos e atitudes.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

1. O cliente precisa estar sempre disponível.

2. Utiliza metáforas no projeto, ou seja, aproxima ao máximo a funcionalidade do incremento à área a que se aplica ou processo a que se destina apoiar. Ex.: sistema de controle de ponto- gera dados para a folha de pagamentos; reserva de carro – incremento para solicitação de carro reserva em uma seguradora que está vinculada a uma apólice de seguro.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

3. Planejamento do jogo com uso de histórias ou use stories, em que há uma reunião utilizando recursos para definição de textos claros da funcionalidade do incremento, em que pode ser utilizada a notação UML para a definição e especificação das regras de negócios. Isso permite inclusive a identificação das atividades que deverão ser realizadas para a entrega das histórias e o seu acompanhamento de perto no cronograma de desenvolvimento do projeto.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

4. Utiliza-se de pequenas versões, também conhecidas como *small releases*. Essa abordagem foca no envio ao cliente das versões que são concluídas, de forma que possa validar a funcionalidade, o incremento realizado.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

5. Também vale- se dos testes de aceitação (*acceptance tests*), que foca o requisito levantado inicialmente e a verificação se a versão atendeu à solicitação real do cliente.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

6. *The first design* ou primeiros testes, aplica-se o conceito de realização de testes por funcionalidade, ou seja, unitários. Estes visam atender às regras de negócios que foram solicitadas, dessa maneira, busca-se a redução de erros de programação.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

7. Integração contínua ou, *continuous integration*, indica que a cada incremento realizado deve ser feito um teste unitário da funcionalidade, até que se obtenha a totalidade da integração.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

8. Simplicidade ou conhecida como *Simple Desing*, já mencionada como sendo interligada à programação que visa facilitar a utilização do sistema, ou seja, tornar fácil ou ao menos amigável a aplicação ao usuário.

Extreme Programming

- **Quadro 1:** Boas práticas de XP

9. Refatoração ou *refactoring*, esta é uma premissa que considera fortemente aspectos de qualidade e melhoria contínua do código do programa.

SCRUM

- Identificação das funcionalidades do projeto (*Product Backlog*).
- Elaboração de uma lista de tarefas chamada de *Sprint Backlog*.
- Valoriza a prática de reuniões diárias para alinhamento de informações entre as equipes e clientes.
- A priorização da solicitação para a sua entrega integral ou seja, produto ou funcionalidade concluída (*Potentially Shippable Product Increment*).

Reunião inicial de planejamento

- ***Sprint Planning Meeting***
 - priorizará as entregas possíveis de serem realizadas no período de duas a quatro semanas.

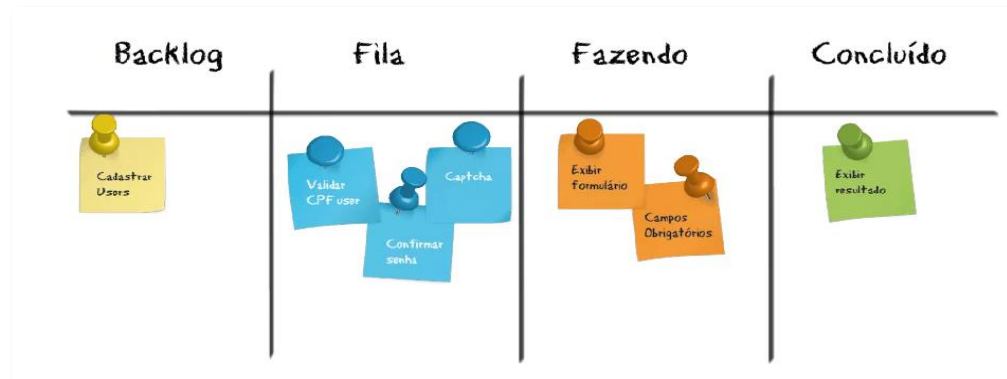


Figura 3: Kanban, quadro de tarefas.

Fonte: BRQ, Metodologias ágeis. Disponível em:
<<http://www.brq.com/metodologias-ageis>>. Acesso em: 2 maio 2016.

Burn Down Chart

- É elaborado para controle das atividades que não devem ultrapassar o período de 24 horas de desenvolvimento.
- Considera basicamente a quantidade de tarefas do *Sprint* pelos dias correspondentes ao ciclo.

Burn Down Chart

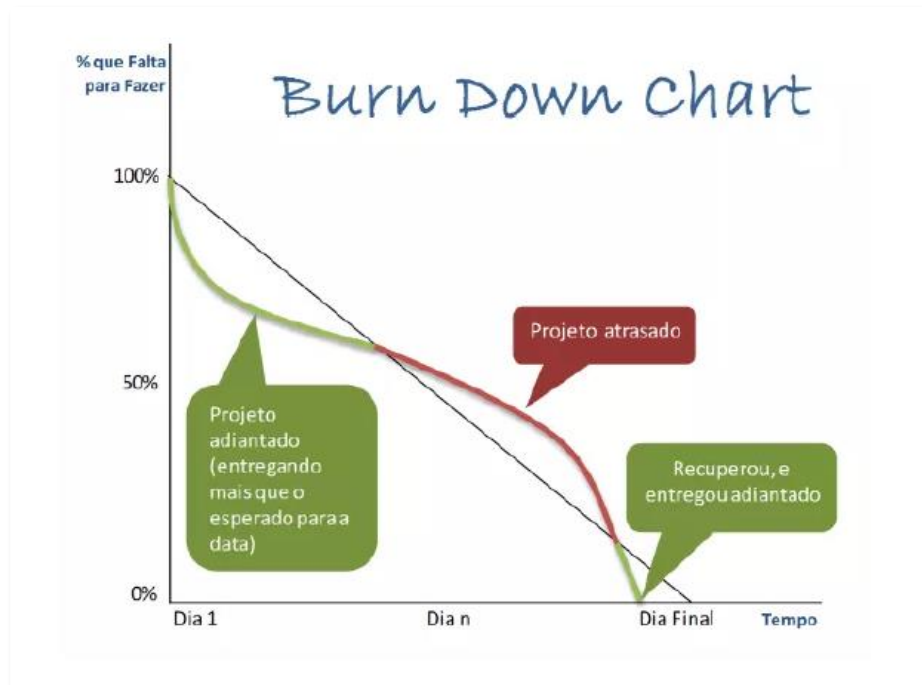


Figura 4: Gráfico de acompanhamento de atividades no *Sprint*.

Fonte: BRQ, Metodologias ágeis. Disponível em:
<<http://www.brq.com/metodologias-ageis>>. Acesso em: 2 maio 2016.

Estrutura da equipe

- ***Product Owner***

- atua no projeto diretamente com o acompanhamento do retorno financeiro do produto (ROI) realizando as estimativas necessárias;
- prioriza as tarefas de acordo com o tempo disponível e tem autonomia para alterar os requisitos de um ciclo (*Sprint*).
- aceita ou não o resultado apresentado pela equipe, controla as entregas.

Estrutura da equipe

- ***Scrum Master***
 - atua mais fortemente com a equipe de análise e desenvolvimento, do ***Scrum Team***.
 - é responsável por garantir a integração da equipe e a sua produtividade.
 - realiza a interface com outras áreas da empresa ou do cliente, para obter mais informações sobre os requisitos.

Estrutura da equipe

- *Scrum Team.*
- **Time de análise, desenvolvimento e implantação dos projetos.**
- Investigue outras metodologias ágeis como: **Rush, PMI-ACP, BDD.**



Obrigada!

Bons estudos.

Referências bibliográficas

SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

PIMENTEL, Manoel. DevMedia. Extreme Programming- conceitos e práticas. Disponível em:

<<http://www.devmedia.com.br/extremeprogrammingconceitosepraticas/1498>>. Acesso em: 02 mai. 2016.

BRQ, Metodologias ágeis. Disponível em: <<http://www.brq.com/metodologias-ageis>>. Acesso em: 02 mai. 2016.

