**Texto base**

**9**

**Abordagens de Desenvolvimento de Software**

Prof. João de Deus Freire Junior

***Resumo***

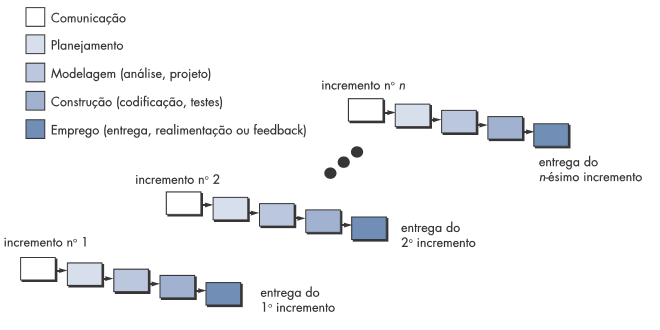
*A Engenharia de Software foi criada com o intuito de resolver ou minimizar os impactos da crise de software. Ela é o emprego de boas práticas no desenvolvimento de software A fim de nos ajudar a aplicar estas melhores práticas podemos utilizar uma das abordagens de desenvolvimento de software que podem nos ajudar a desenvolver produtos de software com qualidade, dentro de prazos e custos previstos que satisfaçam as expectativas dos clientes. Nesta aula, vamos discutir as diferentes abordagens, suas aplicações e diferenças.*

**1.1. Introdução**

O que podemos desenvolver software de forma mais dinâmica? O que é modelo incremental? Quais são as principais abordagens de desenvolvimento de software? Qual a abordagem é a mais recomendada para cada situação? Todas essas perguntas serão respondidas nesta aula. O entendimento das abordagens de desenvolvimento de software, suas principais características e suas aplicações é extremamente importante para que os profissionais de tecnologia da informação tenham competência de saber qual aplicar para suas próprias necessidades.

**1.2. Modelo de Processo Incremental**

No modelo incremental, o sistema é entregue ao cliente em incrementos e cada incremento fornece parte da funcionalidade. Os requisitos são priorizados e os requisitos de prioridade mais alta são incluídos nos incrementos iniciais. Uma vez que o desenvolvimento de um incremento é iniciado, os requisitos são congelados e os requisitos para os incrementos posteriores podem continuar evoluindo e incluir requisitos já implementados. Esse modelo combina elementos dos fluxos de processos lineares e paralelos. Conforme exibido na figura 1.1, o modelo incremental aplica sequências lineares, de forma escalonada, à medida que o tempo vai avançando. Cada sequência linear gera “incrementais” (entregáveis/aprovados/liberados) do software de maneira similar aos incrementais gerados por um fluxo de processos ágeis. (Pressman, 2011).



**Figura 1.1. Modelo de Processo Incremental**

As vantagens desse modelos são :

1. Incrementos podem ser entregues regularmente ao cliente e, desse modo, a funcionalidade do sistema é disponibilizada mais cedo.
2. Os incrementos iniciais agem como protótipos para elucidar os requisitos para incrementos posteriores do sistema.
3. Menor risco de falha geral do projeto
4. Os serviços de sistema de mais alta prioridade tendem a receber mais testes

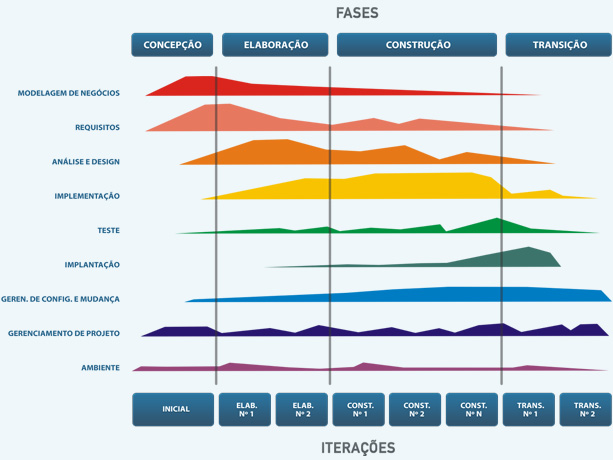
**1.3. A abordagem RUP**

É uma abordagem baseada na UML(Linguagem de Modelagem Unificada). Ela busca cobrir todos os aspectos do desenvolvimento de software. Ela é fortemente focada na documentação do sistema e normalmente é descrita a partir de três perspectivas:

1. Dinâmica: mostra as fases ao longo do tempo.
2. Estática: mostra atividades de processo.
3. Prática: sugere bons princípios e práticas de desenvolvimento.

A abordagem RUP é uma tentativa de aproveitar os melhores recursos e características dos modelos tradicionais de processo de software, mas caracterizando-os de modo a implementar muitas das melhores práticas do desenvolvimento ágil de software. O RUP reconhece a importância da comunicação com o cliente e de métodos sequenciais para descrever a visão do cliente sobre um sistema. Ele enfatiza o importante papel da arquitetura de software e ajuda o arquiteto a manter o foco nas metas corretas, tais como confiabilidade e reutilização. Ele propõe um fluxo de processo iterativo e incremental, proporcionando a sensação evolucionária que é essencial no desenvolvimento de software moderno. (Pressman, 2011)

Segue uma representação gráfica dessa abordagem:



**Figura 1.2. A abordagem RUP - Fases e Atividades**

**1.3.1. Fases do RUP**

* Concepção
  + Estabelecer o domínio de negócio para o sistema
* Elaboração
  + Desenvolver um entendimento do domínio do problema e a arquitetura do sistema
* Construção
  + Projeto, programação e teste de sistema
* Transição
  + Implantar o sistema no seu ambiente operacional

**1.3.2. Boas Práticas do RUP**

* Desenvolver o software iterativamente
* Gerenciar requisitos
* Usar arquiteturas baseadas em componentes
* Modelar o software visualmente
* Verificar a qualidade de software
* Controlar as mudanças do software

**1.4. Manifesto Ágil**

O Manifesto Ágil se trata de um documento que foi assinado e divulgado em 2001, por Kent Beck e outros dezesseis renomados desenvolvedores, autores e consultores da área de software (batizados de “Agile Alliance”- “Aliança dos Ágeis”) assinaram o “Manifesto para o Desenvolvimento Ágil de Software” (“Manifesto for Agile Software Development”). (Pressman, 2011)

O documento se inicia da seguinte maneira:

Desenvolvendo e ajudando outros a desenvolver software, estamos desvendando formas melhores de desenvolvimento. Por meio deste trabalho passamos a valorizar:

* Indivíduos e interações acima de processos e ferramentas
* Software operacional acima de documentação completa
* Colaboração dos clientes acima de negociação contratual
* Respostas a mudanças acima de seguir um plano

Ou seja, embora haja valor nos itens à direita, valorizamos os da esquerda mais ainda.

Normalmente, um manifesto é associado a um movimento político emergente: atacan-

do a velha guarda e sugerindo uma mudança revolucionária (espera-se que para melhor).

De certa forma, é exatamente do que trata o desenvolvimento ágil. Embora as ideias básicas que norteiam o desenvolvimento ágil tenham estado conosco por muitos anos, só há menos de duas décadas que se consolidaram como um “movimento”.

A figura abaixo é uma representação gráfica das principais ideias desse movimento:

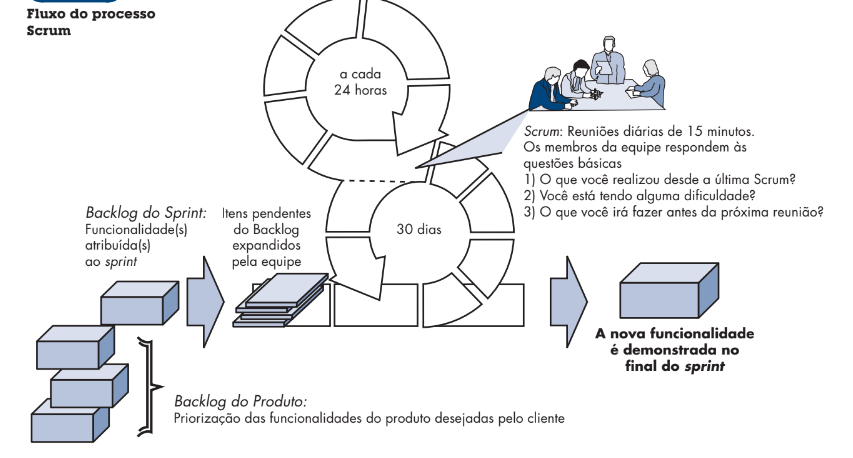


**Figura 1.3. O Manifesto Ágil**

**1.4. A abordagem Scrum**

A palavra Scrum é de origem inglesa e é o nome de uma atividade que ocorre durante a partida de Rugby. Quando utilizada para definir a abordagem de desenvolvimento de software, ela se refere a um método de desenvolvimento ágil de software concebido por Jeff Sutherland e sua equipe de desenvolvimento no início dos anos 1990. Mais recentemente, foram realizados desenvolvimentos adicionais nos métodos gráficos Scrum por Schwaber e Beedle. (Pressman, 2011)

Os princípios da abordagem Scrum são consistentes com o manifesto ágil e são usados para orientar as atividades de desenvolvimento dentro de um processo que incorpora as seguintes atividades estruturais: requisitos, análise, projeto, evolução e entrega. Em cada atividade metodológica, ocorrem tarefas a realizar dentro de um padrão de processo chamado sprint. O trabalho realizado dentro de um sprint (o número de sprints necessários para cada atividade metodológica varia dependendo do tamanho e da complexidade do produto) é adaptado ao problema em questão e definido, e muitas vezes modificado em tempo real, pela equipe Scrum. O fluxo geral do processo Scrum é ilustrado na Figura 1.4. (Pressman, 2011)



**Figura 1.4. Fluxo do processo Scrum**

Em cada ciclo de entrega do Scrum, itens do backlog do produto são retirados e avaliados pela equipe de desenvolvimento junto com os representantes de negócios do produto. Estes itens são entendidos e estimados. Os itens que estiverem dentro da capacidade da equipe para implementação são incluídos na sprint atual e desenvolvidos durante o período da sprint que pode ser de 15, 20 ou até 30 dias. Durante o período da sprint são realizadas reuniões diárias para acompanhamento e alinhamento do trabalho e no fim dela, é realizada uma de entrega dos itens desenvolvidos, chamada de reunião de revisão e outra para avaliar a sprint, chamada de retrospectiva.

As vantagens do Scrum são a adaptabilidade a mudanças devido aos pequenos ciclos de desenvolvimento (sprints), a proximidade do cliente / negócios do desenvolvimento que possibilita maior assertividade na entrega de valor e por fim, desenvolvimento da equipe através das avaliações constantes.

**1.6. Você quer ler?**

Segue uma indicação de estudo complementar. Trata-se de um artigo do site scrum.org que explica o que é o Scrum.

* O que é Scum? Disponível em: <https://www.scrum.org/resources/what-is-scrum>

**1.6. Referências**

* PRESSMAN, R. S.(2011) Engenharia de Software: uma abordagem profissional. 7.ed. Porto Alegre: Bookman, 2016.
* Figuras 1.1, 1.2, 1.3 e 1.4: PRESSMAN, R. S.(2011) Engenharia de Software: uma abordagem profissional. 7.ed. Porto Alegre: Bookman, 2016.