

Análise e Modelagem de Sistemas

Modelos de Processo de Software

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nesta webaula vamos ver alguns Modelos de Processos de Software, disponíveis no mercado, sendo que alguns são bem antigos, mas, mesmo assim, são utilizados em vários projetos. Todos os modelos possuem a finalidade de evitar o caos no desenvolvimento e estabelecer um Fluxo de Trabalho. Na verdade, tudo depende do tipo de software a ser produzido, combinado com as expectativas do cliente.

Modelos de Processo de Software

Durante a fase de desenvolvimento de um software existem muitas tarefas, que devem ser distribuídas entre os membros da equipe. Sommerville (2011) destaca que dentre as atividades do Processo de Software estão: o estudo da viabilidade, a análise dos requisitos, a especificação, a arquitetura de software, implementação (codificação), testes, documentação, suporte e treinamento e manutenção.

Para o gerenciamento das atividades de Processo de Software são utilizados os **Modelos de Processos de Software**.

Modelo de Processo de Software é um guia exclusivo para as atividades da Engenharia de Software, definindo um fluxo de todas as atividades, ações e tarefas, o nível de interação entre as atividades, os artefatos que serão produzidos e a organização do trabalho que deve ser realizado (Pressman, 2016).

Existem diversos Modelos de Processos de Softwares que possuem características diferentes. Destacam-se os seguintes: **Modelos Prescritivos, Modelos de Processos Especializados e Modelos de Desenvolvimento Ágil**.

Modelo de Processo Prescritivo

Também conhecido como Modelos de Processos Tradicionais. Esse tipo de modelo prescreve os relacionamentos, ou seja, como os elementos dos processos são interligados, com a finalidade de estruturar e ordenar o desenvolvimento de um software.

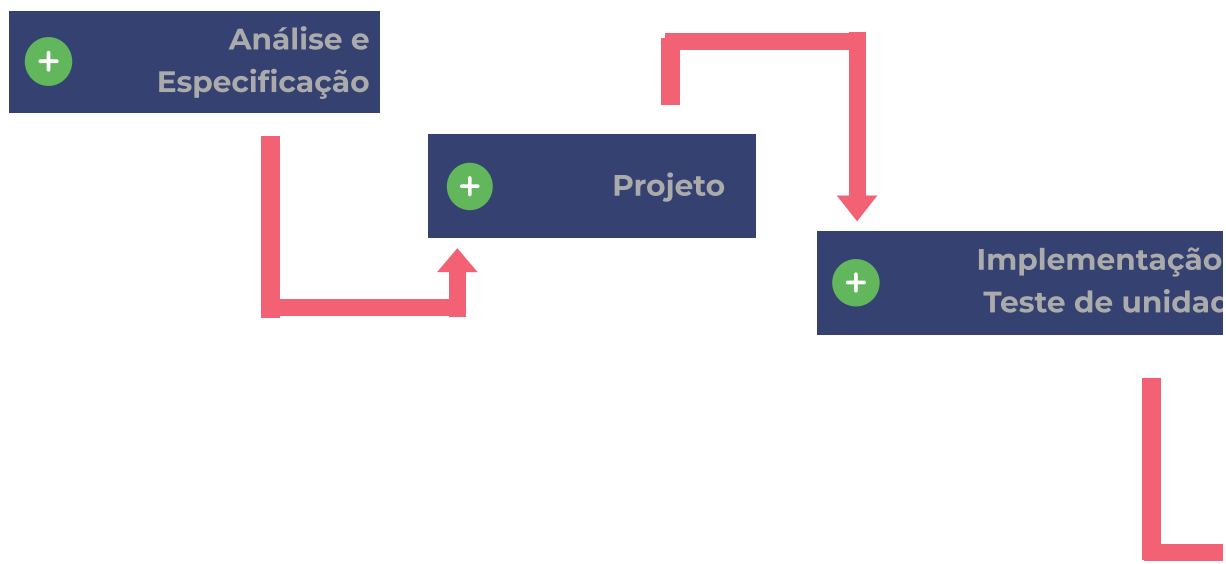
As tarefas ocorrem de forma sequencial, com diretrizes bem definidas, uma vez que indicam uma série de atividades metodológicas, ações, tarefas, artefatos, garantias de qualidade e mecanismos de controle de mudanças para cada projeto. Para cada Modelo de Processo Prescritivo também é indicado um Fluxo de Processo (Pressman, 2016).

Alguns dos Modelos de Processos Prescritivos são os seguintes: **Modelo Cascata, Modelo Incremental, Modelos Evolucionários** (divididos em: Prototipação e Espiral) e **Modelos Concorrentes**.

Modelo Cascata

Conhecido como Ciclo de Vida Clássico de um Sistema ou abordagem *Top-down*, o modelo cascata possui enfoque sistemático e sequencial dos processos, cada fase é iniciada somente após a conclusão da fase anterior.

MODELO CASCAT



Para visualizar o vídeo, acesse seu material digital.

Fonte: adaptado de Sommerville (2011).

O modelo em cascata é considerado o modelo mais tradicional e simples, com especificação das atividades de forma clara, além de ser uma base para modelos que surgiram posteriormente e de fácil gerenciamento. Todavia, o desenvolvimento de um software pode se estender ao longo de meses, dependendo da complexidade do projeto, uma vez que as tarefas são realizadas de forma sequencial e o atraso em uma das etapas reflete nas demais.

Modelo Incremental

É um modelo iterativo que visa, a partir de requisitos iniciais, criar pequenas versões do software, que vão sendo entregues ao cliente, e posteriormente expandir o software em novas versões até o sistema ideal ser totalmente construído (Sommerville, 2011). Nesse modelo, uma versão é um incremento, e cada incremento (ou versão) incorpora parte da funcionalidade requisitada pelo cliente.

No modelo incremental ao invés do cliente receber o software em uma única entrega, ele receberá “pedaços” do software (versões), a cada incremento, até que o Software seja desenvolvido por completo. Esses “pedaços” são módulos que acrescentam, ou melhoram as funcionalidades do sistema. Cada incremento (entrega) é lançado como uma nova versão, do software, até que se atinja a versão final.



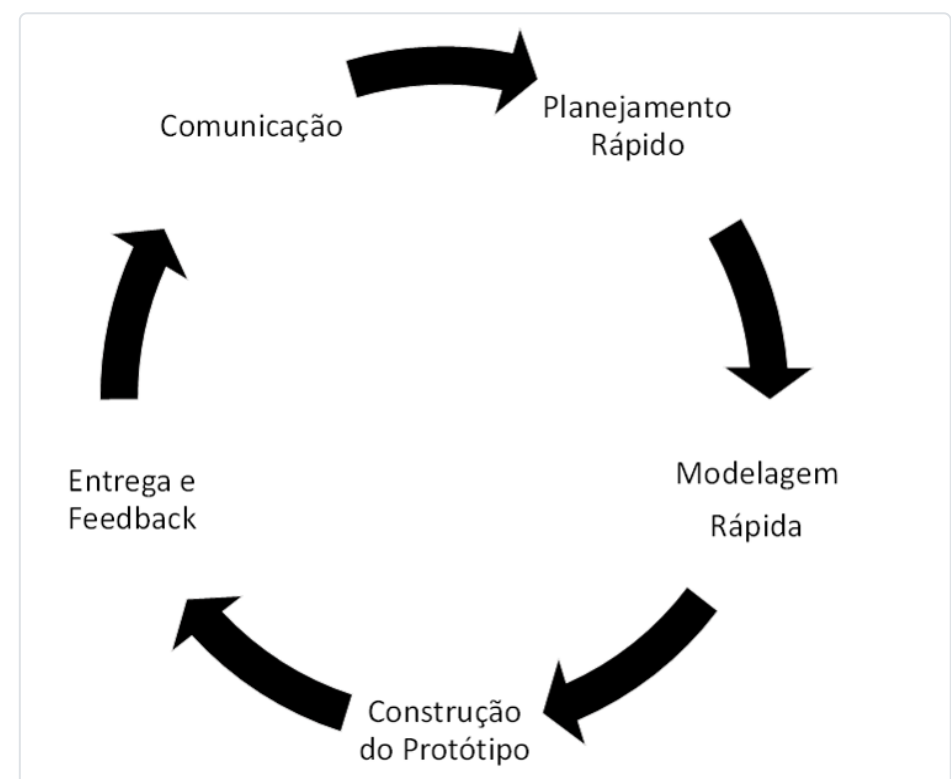
Fonte: adaptada de Sommerville (2011, p. 22).

Modelo Evolucionário

Esse modelo produz uma versão cada vez mais completa do software. Esse tipo de modelo é iterativo e evolui ao longo do tempo, o que se alinha perfeitamente a um projeto de Software, pois os requisitos do negócio e do produto não são estáveis, eles mudam (evoluem) e o Software pode ser desenvolvido pensando nesta evolução (Pressman, 2016). No Modelo de Processo Evolucionário aparecem dois Modelos: Prototipação e Espiral.

O Modelo de Prototipação, começa a partir da Comunicação, identificando quais são os objetivos e as funcionalidades do software. No Planejamento Rápido são determinados os requisitos que serão Modelados (e quais serão “deixados” para serem implementados mais tarde) e que já podem ser Modelados e Construídos. Após a Entrega, o cliente dá um Feedback e a equipe de desenvolvimento irá aprimorar os requisitos.

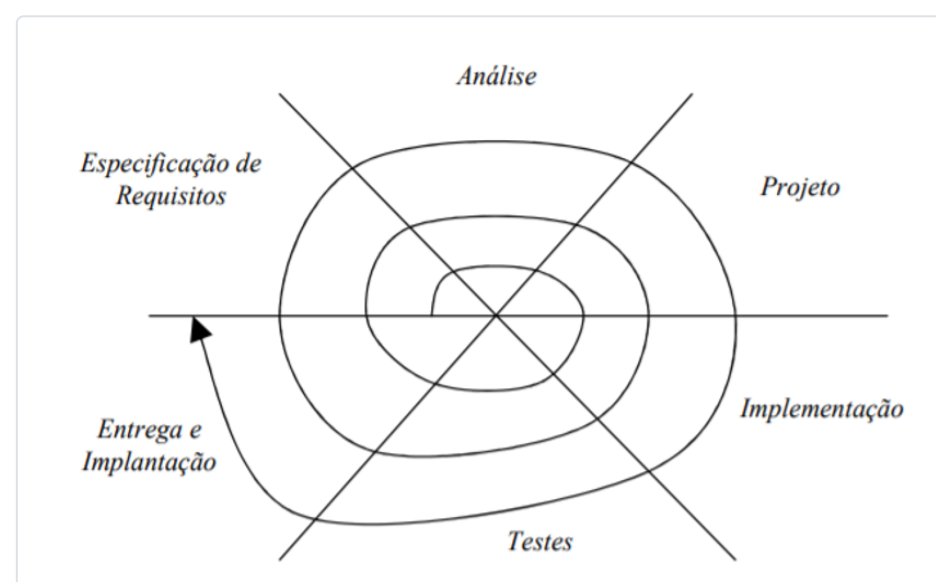
Modelo de Prototipação



Fonte: Pressman (2016, p. 45).

[Saiba Mais](#)

Modelo Espiral



Fonte: Falbo (2012, p. 14).

O Modelo Espiral (um tipo do Modelo Evolutivo) é iterativo como a prototipação, mas utiliza os aspectos sistemáticos e controlados do Modelo Cascata (Pressman, 2016). O objetivo do Modelo Espiral é fornecer um rápido desenvolvimento de versão, que a cada ciclo possa gerar versões mais completas.

Modelos Concorrentes

São representados graficamente por uma série de tarefas e técnicas maiores e estados associados a elas e que são utilizados como um paradigma para o desenvolvimento de aplicações Cliente/Servidor. Os Modelos Concorrentes são utilizados em projetos que envolvem diferentes equipes de desenvolvimento e, conforme Pressman (2016), os planos de projeto devem ser considerados documentos vivos e a evolução de cada processo deve ser avaliada com frequência e revisada levando em consideração as alterações. Os Modelos de Processos

Concorrentes podem ser aplicados a diversos tipos de desenvolvimento de softwares e, diferentemente do Modelo Cascata, ele não segue uma sequência de atividades, mas estabelece uma rede de atividades que se movimentam de uma atividade para outra.

Modelo de Processo Especializado

Esses modelos utilizam muitas das características de um ou mais Modelos de Processos Prescritivos e são utilizados quando existe a necessidade de uma abordagem mais especializada de Engenharia de Software. Conforme Pressman (2016), eles são:

Modelo baseado em componentes	▼
São utilizados em projetos de Software de Prateleira (Software de Linha), compreende aplicações de componentes de Software previamente empacotados (e vendidos em partes ou completo). São desenvolvidos para poderem ser reutilizados em outros projetos. Um componente é uma parte do independente do Software e pode ser trocado ou alterado.	
Modelo de métodos formais	▼
Compreende um conjunto de atividades que levam à especificação matemática formal do Software, fornecendo mecanismos para a descoberta e a eliminação de muitos problemas como a ambiguidade, incompletude e inconsistência. Servem como base para fazer a averiguação do código de programação com o objetivo de descobrir erros (que passariam despercebidos).	
Desenvolvimento de software orientado a aspecto	▼
Fornece um processo e abordagem metodológica para definir, especificar, projetar e construir aspectos. O código do Software é separado de acordo com sua importância (classes orientadas a objetos é um exemplo de aspecto) e os requisitos são modelados transcendendo várias funcionalidades do Sistema.	
Modelo de processo unificado	▼
Conhecida também como RUP – Rational Unified Process) aproveita as características dos Modelos de Processos tradicionais (Prescritivos), mas implementa alguns princípios da metodologia Ágil (abordada mais adiante nesta seção); é considerado um Modelo iterativo e incremental.	
Modelos de processos pessoal e de equipe	▼
O cerne do desenvolvimento de Software está diretamente ligado a toda equipe de desenvolvimento. No Modelo de Processo de Software Pessoal é enfatizada a medição pessoal do que foi produzido (do artefato gerado e a qualidade resultante). O Modelo de Processo de Software de Equipe objetiva a criação de uma equipe autogerida, que se organiza por si mesma com a finalidade de produzir um Software com alto padrão de qualidade.	

Modelo de Desenvolvimento Ágil

O Desenvolvimento Ágil procura resolver alguns problemas da Engenharia de Software, oferecendo benefícios importantes. As etapas de Levantamento de Requisitos, Análise e Projeto são muito demoradas e, de acordo com Pressman (2016), o Desenvolvimento Ágil é uma resposta ao rápido desenvolvimento de software (os clientes querem ver resultados rapidamente) e aplicativos (onde novas funcionalidades são agregadas, na medida que surgem novas ideias ou necessidades), tornando desta maneira o desenvolvimento mais flexível e atendendo as necessidades do cliente com mais rapidez.

O princípio do Desenvolvimento Ágil é focado nas entregas, priorizando também a comunicação entre os envolvidos de forma ativa e contínua para realizar entregas incrementais (procurando a satisfação do cliente). (Pressman, 2016).

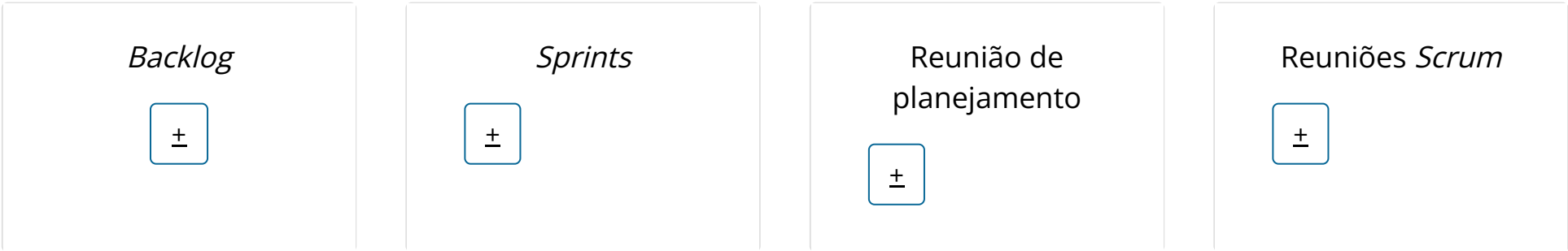
Segundo Pressman (2016), o Processo de Desenvolvimento Ágil visa reduzir drasticamente a documentação, tornando o processo de desenvolvimento flexível e reduzindo a burocracia (presente em outros Modelos de Processos de Softwares). Nesse universo, dois Métodos Ágeis se destacam: **XP (Extreme Programming)** e **Scrum**.

Método (ou metodologia) XP

O Feedback é constante, a abordagem de desenvolvimento é incremental e a comunicação entre os envolvidos é primordial. Quatro atividades metodológicas precisam ser seguidas: o Planejamento, o Projeto, a Codificação e os Testes; o desenvolvimento do Software deve ser padronizado e com o trabalho sendo realizado em pares e o cliente (um representante) deve estar sempre presente para esclarecer dúvidas (ele faz parte da equipe de desenvolvimento).

Metodologia Scrum

Determina um processo de desenvolvimento iterativo e incremental, e pode ser utilizado em processos gerenciais. Esse método define um conjunto de regras e práticas de gestão, para alcançar o sucesso dos projetos como, por exemplo, o trabalho em equipe e comunicação melhorada. O Scrum possui as seguintes atividades metodológicas: Requisitos, Análise, Projeto, Evolução e Entrega; e em cada atividade ocorrem as seguintes tarefas principais:



- Quando o projeto inicializa, são definidas as ideias e funcionalidades iniciais do produto a ser desenvolvido, estas ideias são chamadas **Histórias** e o conjunto de todas as Histórias forma o **Product Backlog**.
- As reuniões são conduzidas pelo Scrum Master que conduz o processo e realiza avaliações das respostas de cada integrante da equipe, detectando de forma precoce eventuais problemas, como atrasos ou dificuldade de entendimento de algum requisito.
- No término da **Sprint** os requisitos são concluídos e o funcionamento é avaliado, melhorando o processo para a **Sprint** sequencial. Cada Sprint se encerra com um incremento ao produto (ou **Product Backlog**).

Na metodologia Scrum sempre é montado um quadro (board) para acompanhar as tarefas. No quadro são inseridas as atividades Stories, To Do, In Progress, Testing e Done.

Esse quadro pode ser adaptado para a realidade de cada equipe de desenvolvimento.



Fonte: Shutterstock.

Existem ainda outros métodos de Desenvolvimento Ágil, Pressman (2016) lista os seguintes: Método de Desenvolvimento de Sistemas Dinâmicos (DSDM), Modelagem Ágil e Processo Unificado Ágil. Todos os métodos ágeis enfatizam a colaboração humana e a auto-organização como elementos chaves.

Vimos os Modelos de Processos de Software e seus diferentes tipos: Modelos Prescritivos, Modelos Especializados e Modelos Ágeis. Todos os Modelos apresentados são amplamente utilizados nas empresas de desenvolvimento de software. São fatores determinantes para a escolha do tipo de modelo a ser usado: o tipo de software que deverá ser produzido e a experiência dos Analistas de Sistemas (envolvidos nos projetos).

Pesquise mais!

Para conhecer mais sobre o Processo de Software para Metodologias Ágeis, consulte o capítulo 1: Processos de Software, no livro: Engenharia de software [recurso eletrônico] de VETTORAZZO, A. de S. Revisão técnica: Fábio Josende Paz. Porto Alegre: SAGAH, 2018, p. 12-23. O livro está disponível na Biblioteca Virtual.



Para visualizar o vídeo, acesse seu material digital.