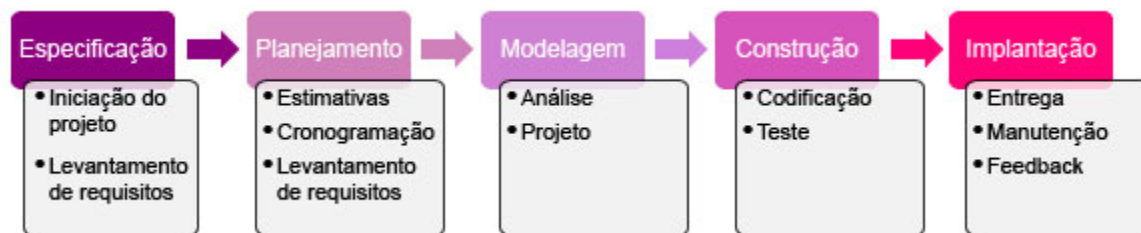


Modelos de desenvolvimento: cascata e evolucionário (exploratório e prototipação)

APRESENTAR OS CONCEITOS E CARACTERÍSTICAS DOS MODELOS DE PROCESSO CASCATA E EVOLUCIONÁRIO (EXPLORATÓRIO E PROTOTIPAÇÃO).

Modelo cascata

No modelo em cascata, também conhecido como ciclo de vida clássico, sequencial linear ou waterfall, o processo de desenvolvimento de software é visto como uma abordagem sistemática e sequencial que começa com a especificação dos requisitos do cliente. Assim, progride seguindo as etapas de planejamento, modelagem, construção e implantação do sistema, conforme ilustra a figura adiante, culminando na manutenção progressiva do produto entregue.



Legenda: MODELO EM CASCATA ([HTTP://WWW.SHUTTERSTOCK.COM](http://www.shutterstock.com))

Este modelo é o paradigma mais antigo da engenharia de software e é bastante criticado em função dos problemas encontrados nos projetos em que é aplicado.

A realidade tem mostrado que num projeto raramente segue-se o fluxo sequencial que o modelo propõe, gerando problemas futuros que oneram os custos e prazos. Uma das causas mais comuns deste problema é a dificuldade do cliente em declarar de forma clara todas as suas necessidades e expectativas, ou seja, de definir todos os requisitos inicialmente.

O foco incorreto ou não claro pode gerar uma distorção a qual reflete diretamente na percepção de qualidade por parte do próprio cliente. Isso pode levar a entregas parciais do produto – o qual exige que o cliente tenha muita paciência durante os aceites parciais formalizados em um documento de

encerramento do projeto com observações no campo restrições "entrega parcial de projeto", o qual contém detalhes quanto à aceitação condicional do projeto por parte do cliente.

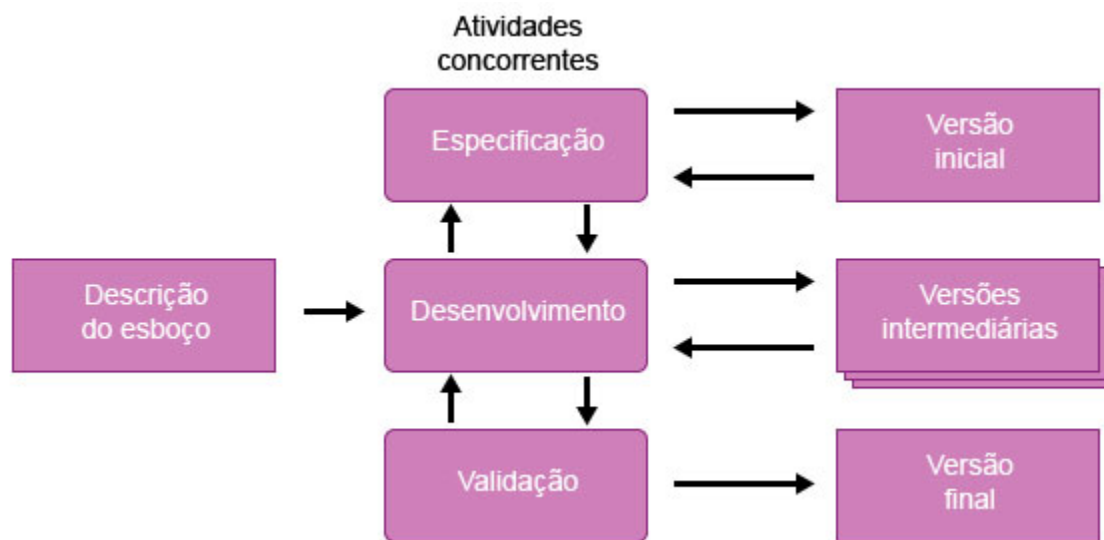
Os trabalhos de desenvolvimento de software atuais seguem ritmos muito rápidos, sujeitos a diversas modificações, o que torna o modelo em cascata inadequado para estes tipos de projeto.

Mas, cumpre ressaltar que embora o modelo em cascata ou ciclo de vida clássico tenha fragilidades, ele é significativamente melhor do que uma abordagem casual para o desenvolvimento de software.

Modelo evolutivo

Tem como base a ideia de desenvolver uma implementação inicial, expor o resultado ao comentário do usuário e fazer seu aprimoramento por meio de muitas versões, até que tenha sido desenvolvido. A especificação, desenvolvimento e validação são executados concorrentemente para gerar um retorno rápido. Podem ser:

- Exploratório: tem como objetivo trabalhar com o cliente a fim de explorar seus requisitos e entregar um sistema final. São feitas partes iniciais e acrescentadas novas de acordo com o desenvolvimento, conforme ilustrado na figura a seguir.

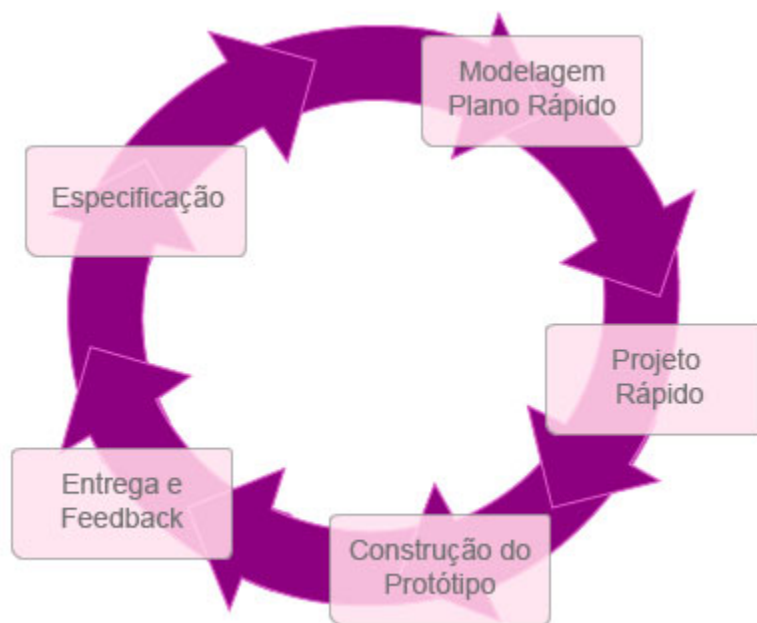


- Protótipos descartáveis: É um modelo de processo que possibilita que o desenvolvedor crie um modelo do software que deve ser construído. Idealmente, o modelo (protótipo) serve como um mecanismo para identificar os requisitos de software. É apropriado para quando o cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento e saída com detalhes.

Envolve o desenvolvimento de uma versão inicial do sistema baseada no atendimento dos requisitos ainda pouco definidos. Este processo permite a descoberta de falhas difíceis de serem encontradas na comunicação verbal, servindo de apoio à fase de levantamento de requisitos e prevenindo as possíveis

falhas no sistema.

Conforme descrito por Pressman (2001), o objetivo principal de um protótipo é simular a aparência e funcionalidade do software permitindo que os clientes, analistas, desenvolvedores e gerentes compreendam plenamente os requisitos do sistema interagindo, avaliando, alterando e aprovando as características mais relevantes que o produto deva ter. A figura a seguir ilustra este processo.



Legenda: MODELO DE PROTOTIPAÇÃO (HTTP://WWW.SHUTTERSTOCK.COM)

Certamente a redução de custos no desenvolvimento é um dos grandes ganhos da prototipação, pois envolve diretamente o usuário final permitindo um desenvolvimento mais próximo dos desejos do cliente, priorizando a facilidade de uso. Assim, pode-se obter um nível de satisfação maior em função do menor número de erros ou falhas de desenvolvimento comuns na passagem de informação entre o analista (que fez o levantamento de requisitos) e o desenvolvedor (equipe de desenvolvimento).

Alguns problemas podem ocorrer na aplicação desse modelo. Um deles é que o cliente não sabe que o software que ele vê não considerou, durante o desenvolvimento, a qualidade global e a manutenção a longo prazo. Ele pode, também, não aceitar facilmente a ideia de que a versão final do software está sendo construída e tentar forçar a utilização do protótipo como produto final.

Outro risco é que o desenvolvedor, frequentemente, faça uma implementação comprometida (utilizando partes de programas existentes, geradores de relatórios e de janelas) com o objetivo de produzir rapidamente um protótipo executável. Depois de um tempo ele se familiariza com essas escolhas, e pode se esquecer que elas não são apropriadas para o produto final.

Ainda que possam ocorrer problemas, a prototipação é um modelo eficiente. A chave para seu sucesso é definir as regras do jogo logo no começo.

O cliente e o desenvolvedor devem concordar que o protótipo seja construído para servir como um mecanismo a fim de se definir os requisitos do projeto.

Agora que você já estudou esta aula, resolva os exercícios e verifique seu conhecimento. Caso fique alguma dúvida, leve a questão ao Fórum e divida com seus colegas e professor.

EXERCÍCIO (https://ead.uninove.br/ead/disciplinas/web/_g/pdsoft80_100/a04ex01_pdsoft80_100.htm)

REFERÊNCIA

PRESSMAN, R. S. *Engenharia de software*. 7ª ed. São Paulo: McGraw-Hill, 2010..

SOMMERVILLE, Ian. *Engenharia de software*. São Paulo: Addison-Wesley, 2007.

