Conceitos gerais e principais abordagens de desenvolvimento de software

Metodologias clássicas

Bloco 1

Thiago Salhab Alves

Metodologias clássicas

Objetivos:

- Aprender sobre as metodologias clássicas para desenvolvimento de sistemas.
- Aprender sobre as metodologias cascata, prototipação e baseado em componentes.
- Aprender sobre as metodologias incremental e espiral.

Metodologias clássicas

- Metodologias clássicas foram propostas em um momento que o uso de *mainframes* era dominante e não existiam ferramentas de apoio ao desenvolvimento de software (SBROCCO; MACEDO, 2012).
- Havia uma grande preocupação com planejamento e documentação do projeto antes da sua implementação.
- Os modelos de processo foram propostos para se trazer ordem ao caos que se instaurou na área de desenvolvimento de software (SBROCCO; MACEDO, 2012).

Metodologias clássicas

- Esses modelos proporcionam uma considerável contribuição para a estrutura utilizável no trabalho de engenharia de software (PRESSMAN, 2016).
- Um caminho eficaz a ser seguido pelas equipes de Engenharia de software poderia ser utilizado (PRESSMAN, 2016).
- Um processo de desenvolvimento de software foi proposto.
- Processo de desenvolvimento de software é um conjunto de atividades que produz um produto de software (PRESSMAN, 2016).

Metodologias clássicas

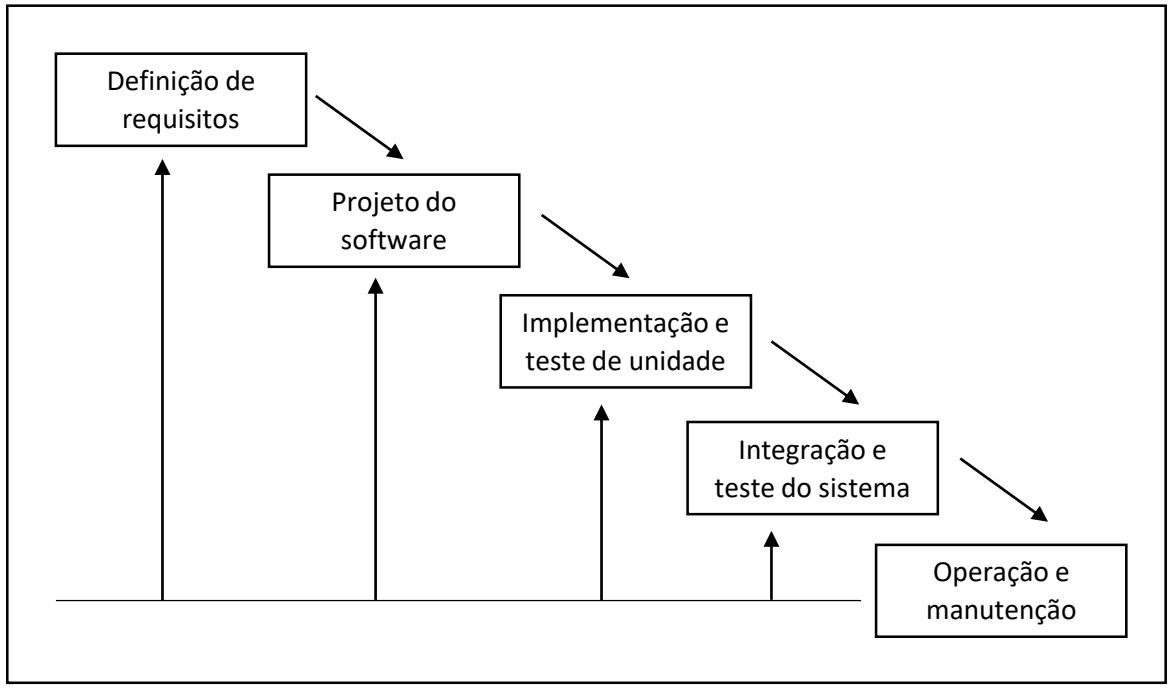
- Um modelo de processo tem por objetivo estruturar e ordenar o desenvolvimento de software (PRESSMAN, 2016).
- O modelo clássico ou sequencial foi o primeiro modelo proposto como processo de desenvolvimento de software (PRESSMAN, 2016).
- Modelo de fácil entendimento, com sequência de etapas bem definidas.
- Cada etapa tem associada ao seu término uma documentação que deve ser aprovada para que se inicie a próxima etapa.

Metodologias clássicas

- O Modelo Cascata, ou Modelo Sequencial Linear, foi publicado em 1970 tendo esse nome porque o processo é visto como um fluir constante para frente (como uma cascata) e aplicado para sistemas de grande porte, conforme Sommerville (2011). Possui as seguintes etapas:
    - Definição de requisitos.
    - Projeto do software.
    - Implementação e teste de unidade.
    - Integração e teste do sistema.
    - Operação e manutenção.

Metodologias clássicas

Figura 1 – Modelo Cascata



Fonte: elaborada pelo autor.

Metodologias clássicas

Vantagens associadas ao modelo cascata (PRESSMAN, 2016):

- O processo de desenvolvimento tem uma ordem sequencial de etapas em que cada etapa deve estar terminada para o início da próxima.
- Maior garantia que os requisitos estão completos e os softwares produzidos correspondem ao que foi especificado.
- Processo de desenvolvimento conduzido de forma disciplinada e como atividades definidas e determinadas a partir de um planejamento.
- Documentação completa do sistema .

Metodologias clássicas

Problemas associados ao modelo cascata (PRESSMAN, 2016):

- Projetos dificilmente seguem o fluxo sequencial.
- Modelo linear é aplicado de forma indireta.
- Cliente tem dificuldade em determinar todas às suas necessidades.
- Cliente só receberá versão final quando o projeto finalizar.
- Cliente tem incertezas que podem prejudicar o andamento do projeto.

Metodologias clássicas

Bloco 2

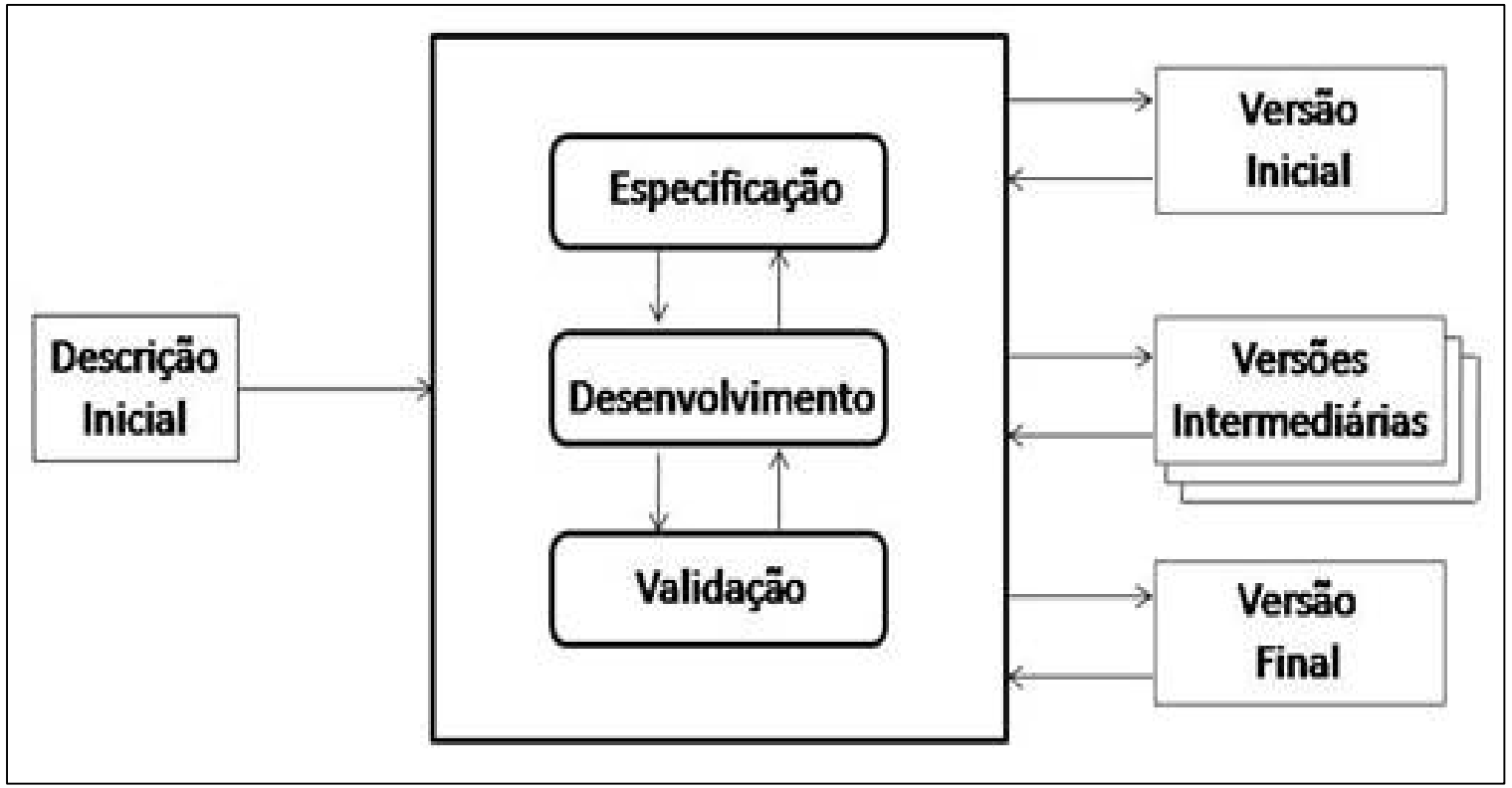
Thiago Salhab Alves

Prototipação

- A prototipação toma por base para o desenvolvimento uma descrição inicial, que é submetida a avaliação do usuário, que aponta suas impressões e permite um refinamento por meio de várias versões até que seja desenvolvido um sistema adequado. Introduzir melhor o que é a prototipação (SOMMERVILLE, 2011).
- A prototipação apresenta dois tipos de desenvolvimento:
    - Exploratório.
    - Prototipação.

Prototipação

Figura 2 – Modelo prototipação



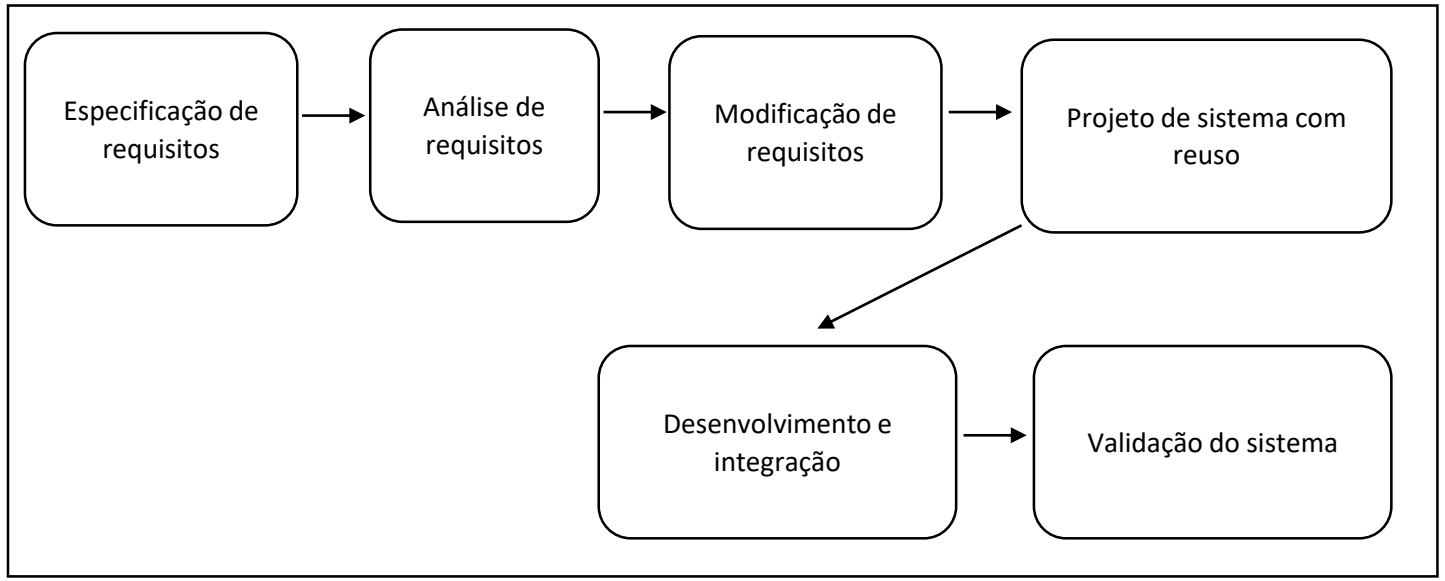
Fonte: elaborada pelo autor.

Modelo baseado em componentes

- O desenvolvimento baseado em componentes, descrito por Sommerville (2011), depende de uma grande base de componentes de softwares reusáveis e e um framework que integre esses componentes.
- Os componentes são sistemas comerciais independentes que podem fornecer funcionalidade específica, como formatação de texto ou cálculo numérico.

Modelo baseado em componentes

Figura 3 – Modelo baseado em componentes



Fonte: elaborada pelo autor.

Metodologias clássicas

Bloco 3

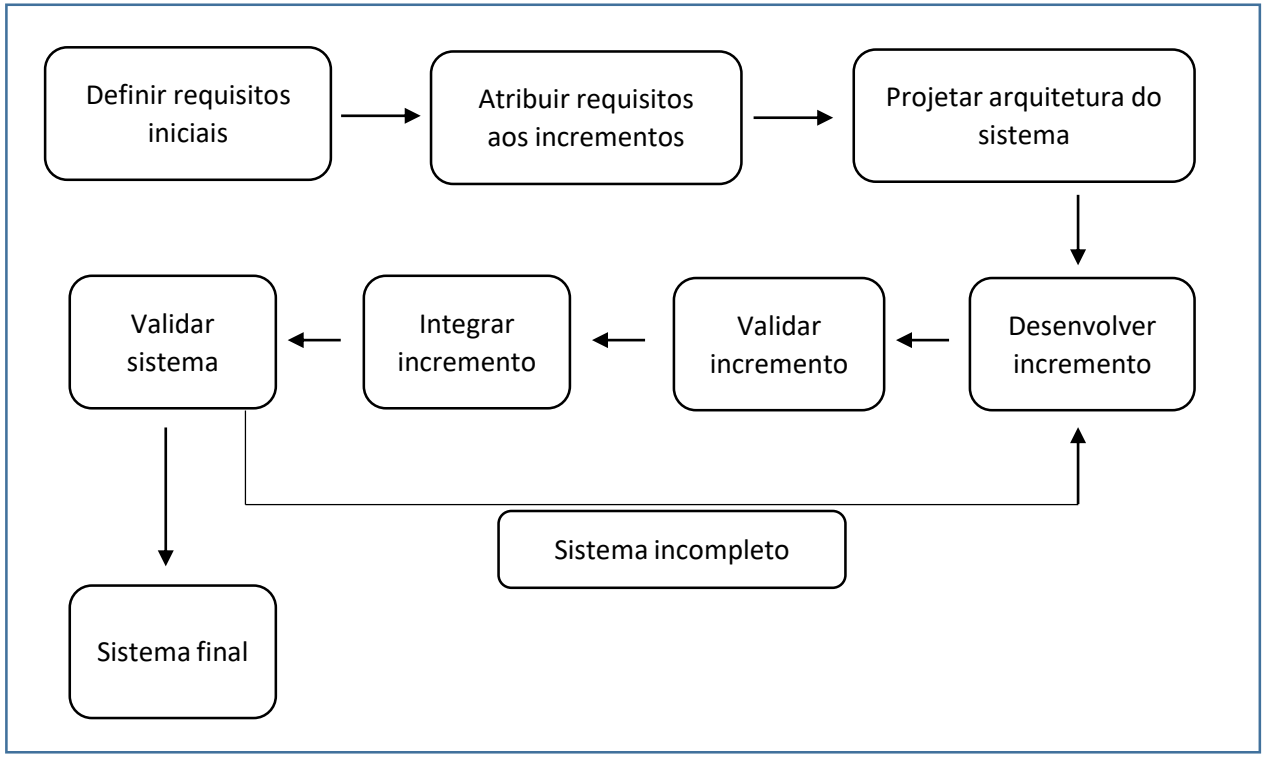
Thiago Salhab Alves

Metodologia incremental

- Para Sommerville (2011), a metodologia incremental é uma abordagem que combina as vantagens do Modelo Cascata.
- O cliente identifica os serviços a serem fornecidos pelo sistema e quais são os mais e menos importantes.
- Os incrementos de entrega são definidos e cada incremento fornece um subconjunto das funcionalidades do sistema.
- Quando o incremento for concluído, o mesmo é entregue e o cliente pode colocar em operação.

Metodologia incremental

Figura 4 – Modelo Incremental



Fonte: elaborada pelo autor.

Metodologia incremental

Vantagens da metodologia incremental (SOMMERVILLE, 2011):

- Os clientes não precisam esperar o sistema ser concluído para poder utilizar.
- Quando os incrementos forem concluídos, já é possível a utilização do software.
- Os incrementos iniciais, podem ser usados como protótipos, assim, o cliente pode ir ganhando experiência a partir de seu uso.
- Menor risco de falha geral do projeto.

Metodologia incremental

Desvantagens da metodologia incremental (SOMMERVILLE, 2011):

- Os incrementos devem ser relativamente pequenos.
- A maior parte dos sistemas requer um conjunto de recursos básicos usados por diferentes partes do sistema.
- Uma variante dessa abordagem incremental é denominada *extreme programming*.

Modelo Espiral

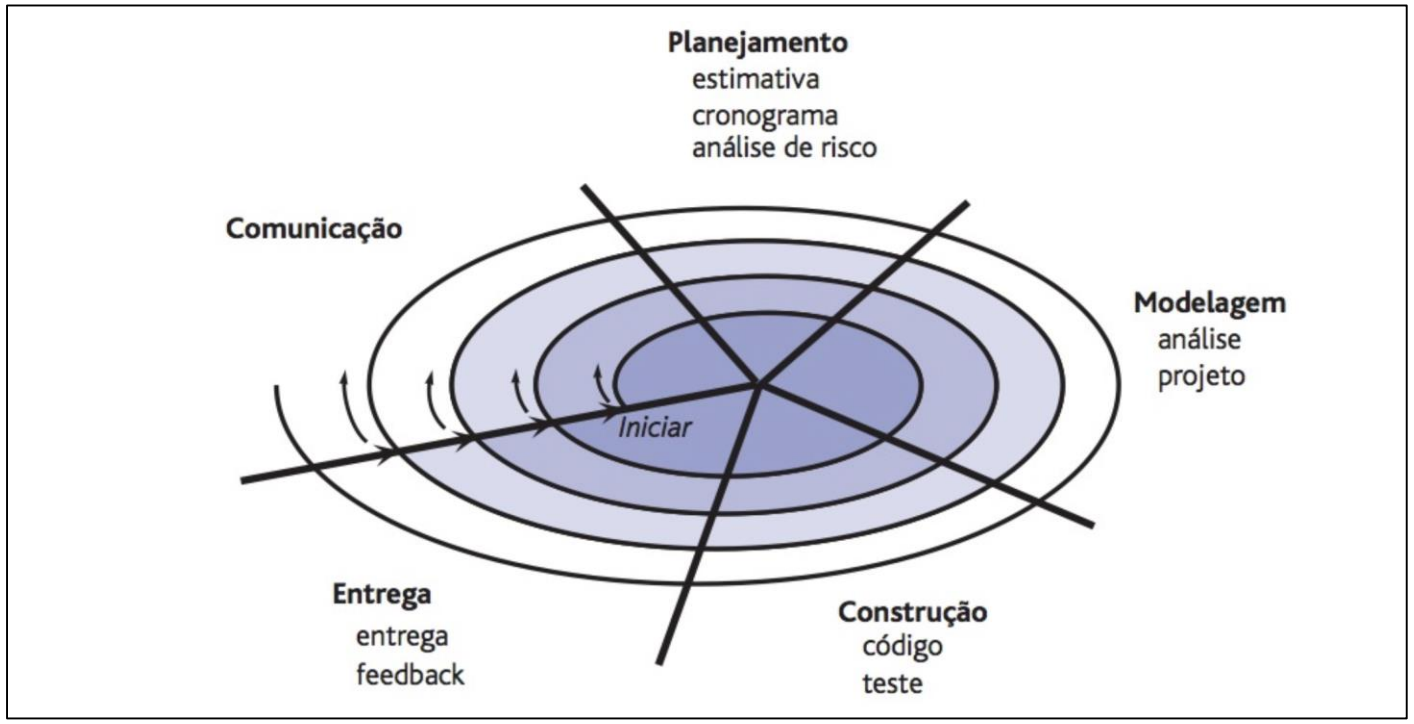
- O Modelo Espiral é um modelo de software evolucionário que une a parte iterativa da prototipação e a parte sistemática e controlada do Modelo Cascata, conforme Paula Filho (2019).
- O software é desenvolvido em uma série de versões evolucionárias, sendo que nas primeiras iterações, a versão se constitui em um modelo ou protótipo e, nas iterações posteriores, produzidas versões cada vez mais completas do sistema que passa pelo processo de engenharia.
- O processo é representado como um espiral, e cada *loop* representa uma fase do processo de software.

Modelo Espiral

- De acordo com Pressman (2016), a principal diferença entre o modelo em espiral e os outros modelos do processo de software é o reconhecimento explícito do risco no modelo em espiral.
- Risco significa simplesmente algo que pode dar errado.
- Diferentemente de outros modelos que terminam quando o software é entregue, o esse modelo pode ser adaptado para ser aplicado ao longo da vida do software.

Modelo Espiral

Figura 5 – Modelo Espiral



Fonte: Pressman (2016, p. 48).

Teoria em Prática

Bloco 4

Thiago Salhab Alves

Reflita sobre a seguinte situação

Você foi promovido a engenheiro de software em sua organização e tem a difícil tarefa de orientar a mudança do atual processo de desenvolvimento de sistemas, com o objetivo de **melhorar os produtos** e a **aceitação por parte dos clientes**. A empresa trabalha com o desenvolvimento de sistemas para **diversos setores comerciais**, o que faz com que os **requisitos mudem constantemente**. Muitos dos sistemas desenvolvidos atualmente **apresentam atrasos** e muitos **erros de funcionamento**. Dessa forma, uma vez que você conheceu vários modelos clássicos de desenvolvimento de sistemas, quais metodologias de desenvolvimento você indicaria para ser aplicado no dia a dia de trabalho da organização? Faça as propostas e apresente-as, em forma de relatório, ao diretor da empresa .

Norte para a resolução...

- Diversas são as metodologias clássicas que podem ser aplicadas pelo engenheiro de software na organização, mas por se tratar de um ambiente com constante mudança nos requisitos e a necessidade de um processo rápido de desenvolvimento, podem ser aplicadas as seguintes metodologias:
    - Prototipação: utilizar o desenvolvimento exploratório ou prototipação;
    - Baseado em componentes: determinar uma base de componentes que possam ser aplicados e agilizar o processo de desenvolvimento.
    - Metodologia incremental: identificar as necessidades mais importantes e que devem ser fornecidos pelo sistema.

Dica do Professor

Bloco 5

Thiago Salhab Alves

Dica do Professor

Sugestão de leitura de artigo:

- SALEEM ABBAS, A.; JEBERSON, W.; KLINSEGA, V. V. Proposed Software Re-engineering Process That Combine Traditional Software Re-engineering Process With Spiral Model. **International Journal of Advanced Research in Computer Science**, [s. l.], v. 4, n. 1, p. 75–83, 2013. Disponível em: http://search.ebscohost.com/login.aspx?direct=true&db=scf&AN=91876775&lang=pt-br&site=ehost-live. Acesso em: 12 maio 2020.

Referências

PAULA FILHO, W. P. **Engenharia de Software:** produtos. 4. ed. Rio de Janeiro: Editora LTC, 2019.

PRESSMAN, R. **Engenharia de Software**. 6. ed. São Paulo: Makron Books, 2006.

PRESSMAN, R. **Engenharia de Software:** uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016.

SBROCCO, J. H. T.; MACEDO, P. C. **Metodologias ágeis:** engenharia de software sob medida. 1. ed. São Paulo: Érica, 2012.

SOMMERVILLE, I. **Engenharia de Software**, 9. ed. Pearson Education do Brasil, 2011.

Bons estudos!