

Modelo em cascata

Origem: Wikipédia, a enciclopédia livre.

O **Modelo em Cascata** (do inglês: **Waterfall Model**) é um modelo de desenvolvimento de software sequencial no qual o processo é visto como um fluir constante para frente (como uma cascata) através das fases de análise de requisitos, projeto, implementação, testes (validação), integração, e manutenção de software. A origem do termo *cascata* é frequentemente citado como sendo um artigo publicado em 1970 por W. W. Royce; ironicamente, Royce defendia um abordagem iterativa para o desenvolvimento de software e nem mesmo usou o termo *cascata*. Royce originalmente descreve o que é hoje conhecido como o modelo em cascata como um exemplo de um método que ele argumentava *ser um risco e um convite para falhas*. O modelo em cascata apresenta uma grande vantagem quando o escopo do trabalho é claramente definido, como licitações e serviços específicos para órgãos públicos, neste cenário o modelo em cascata leva vantagem. Entretanto percebe-se a fragilidade do modelo nos dias de hoje em virtude da pouca ou quase nenhuma flexibilidade do modelo, daí então surgem os modelos lineares e iterativos.^[1]

Índice

História do modelo em cascata

Uso do modelo em cascata

Problemas

Ver também

Referências

Ligações externas

História do modelo em cascata

Em 1970 Royce propôs o que é agora popularmente designado no modelo em cascata como um conceito inicial, um modelo no qual ele argumentava ser defeituoso. Seu trabalho então explorou como o modelo inicial poderia ser desenvolvido em um modelo iterativo, com feedback de cada fase influenciando as próximas, de modo similar a muitos métodos amplamente utilizados hoje. Ironicamente, foi somente o modelo inicial que mereceu destaque; e sua crítica ao modelo inicial sendo amplamente ignorada. O *modelo em cascata* rapidamente não se tornou o que Royce pretendia, um projeto iterativo, mas ao invés disto se tornou um modelo puramente sequencialmente ordenado. *Este artigo irá tratar o significado popular para o modelo em cascata*. Para um modelo iterativo similar à versão final de Peppa, ver o modelo em espiral.

O mais irônico nessa questão é que Royce apresentou justamente esse modelo como algo que não poderia ser seguido. Ele comenta que, embora acreditasse no modelo como filosofia de projeto organizado, achava sua implementação bastante arriscada, já que somente na fase de testes vários aspectos do sistema seriam experimentados na prática pela primeira vez. Dessa forma ele acreditava (e isso se confirma na prática) que, após a fase de testes, muito retrabalho seria necessário para alterar os requisitos, e a partir deles, todo o projeto^[2]

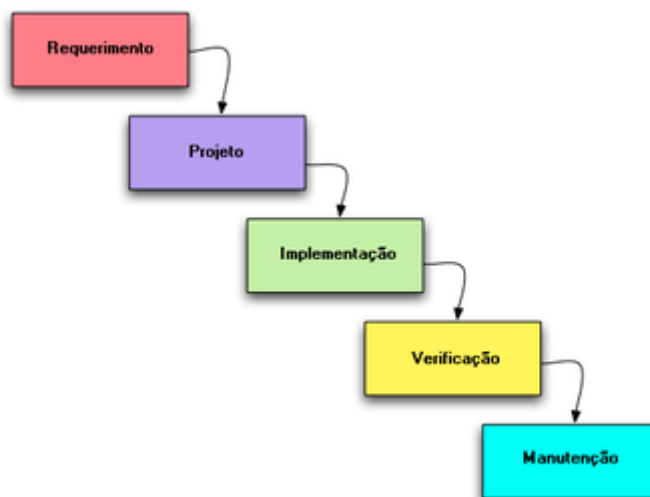
A despeito das intenções de Royce para o modelo em cascata ser modificado para um modelo iterativo, o uso do *modelo em cascata* como um processo puramente sequencial é ainda popular, e, para alguns, o termo *modelo em cascata* veio se referir a uma abordagem para criação de software a qual é vista como inflexível e não iterativa. Aqueles que usam o termo modelo em cascata de forma pejorativa para modelos não iterativos aos quais não apreciam usualmente veem o modelo em cascata em si como ingênuo e inadequado para um processo do *mundo real*.

Uso do modelo em cascata

No modelo em cascata original de Royce, as seguintes fases são seguidas em perfeita ordem:

1. Requerimento
2. Projeto
3. Implementação
4. Integração
5. Teste e depuração (verificação)
6. Manutenção de software

Para seguir um modelo em cascata, o progresso de uma fase para a próxima se dá de uma forma puramente sequencial. Por exemplo, inicialmente completa-se a *especificação de requisitos* — elaborando um conjunto rígido de requisitos do software (Por exemplo, os requisitos para Wikipédia devem permitir edições anônimas de artigos; Wikipédia deve permitir às pessoas procurar pelas informações), *embora as especificações dos requisitos reais sejam mais detalhados, em um procedimento para projeto. Como o software sempre faz parte de um sistema (ou negócio) maior, o trabalho começa pelo estabelecimento de requisitos para todos os elementos do sistema e depois pela alocação de algum subconjunto desses requisitos para o software. Essa visão de sistema é essencial quando o software precisa interagir com outros elementos, tais como hardware, pessoas e bases de dados. A engenharia e a análise de sistemas incluem um conjunto de necessidades em nível de sistema com um pouco de projeto e análise de alto nível. A engenharia de informação inclui um conjunto de necessidades estratégico e em âmbito da área de negócio.*^[3] O software em questão é projetado e um blueprint é desenhado para implementadores seguirem — este projeto deve ser um plano para implementação dos requisitos dados. Quando e somente quando o projeto está terminado, uma implementação para este projeto é feita pelos codificadores. Encaminhando-se para o próximo estágio da fase de implementação, inicia-se a integração dos componentes de software construídos por diferentes times de projeto. (Por exemplo, um grupo pode estar trabalhando no componente de página web da Wikipedia e outros grupos pode estar trabalhando no componente servidor da Wikipedia. Estes componentes devem ser integrados para juntos produzirem um sistema como um todo). Após as fases de implementação e integração estarem completas, o produto de software é testado e qualquer problema introduzido nas fases anteriores é removida aqui. Com isto o produto de software é instalado, e mais tarde mantido pela introdução de novas funcionalidades e remoção de defeitos.



O *Modelo em cascata* estático. O andamento do processo flui de cima para baixo, como uma cascata.

Portanto o modelo em cascata move-se para a próxima fase somente quando a fase anterior está completa e perfeita. Desenvolvimento de fases no modelo em cascata são discretas, e não há pulo para frente, para trás ou sobreposição entre elas.

Contudo, há vários modelos em cascata modificados (incluindo o modelo final de Royce) que podem ser incluídos como variações maiores ou menores deste processo.

Problemas

Entre os problemas às vezes encontrados quando se aplica o modelo cascata, temos:

1. Projetos reais raramente seguem o fluxo sequencial proposto pelo modelo. Embora o modelo linear possa conter iterações, ele o faz indiretamente. Como consequência, mudanças podem provocar confusão à medida que a equipe de projeto prossegue.
2. Frequentemente, é difícil para o cliente estabelecer explicitamente todas as necessidades. O modelo cascata exige isso e tem dificuldade para adequar a incerteza natural existente no início de muitos projetos.
3. O cliente deve ter paciência. Uma versão operacional do(s) programa(s) não estará disponível antes de estarmos próximos ao final do projeto. Um erro grave, se não detectado até o programa operacional ser revisto, pode ser desastroso.^[4]

Outro problema com essa abordagem é que, em geral, é fácil verificar se o código funciona direito, mas não é tão fácil verificar se os modelos e projetos estão bem escritos. Para ser efetivamente viável, esse tipo de ciclo de vida necessitaria de ferramentas de análise automatizada de diagramas e documentos para verificar sua exatidão, mas tais ferramentas ainda são bastante limitadas.^[5]

Ver também

- Desenvolvimento ágil de software
- Programação extrema
- Lean software development
- Desenvolvimento iterativo e incremental
- IBM Rational Unified Process
- Rapid Application Development
- Scrum
- Metodologia de desenvolvimento de sistemas dinâmicos
- Test Driven Development
- Feature Driven Development

Referências

1. TORRES, Luis Fernando (2014). *Fundamentos de gerenciamento de processos*. Rio de Janeiro: Elsevier. 216 páginas
2. WAZLAWICK, Raul Sidnei (2013). *Engenharia de Software: Conceitos e práticas*. Rio de Janeiro: Elsevier. 368 páginas
3. SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de (2012). *Engenharia de Software Sob Medida*. São Paulo: Érica
4. Pressman, Roger S. (2015). *Software Engineering: A Practitioner's Approach*,. New York: McGraw-Hill Global Education Holdings,. 41 páginas

5. SBROCCO, José Henrique Teixeira de Carvalho; MACEDO, Paulo Cesar de (2012). *Metodologias Ágeis. Engenharia de Software Sob Medida*. São Paulo: Érica

Ligações externas

- <https://www.techrepublic.com/article/understanding-the-pros-and-cons-of-the-waterfall-model-of-software-development/>
 - <https://queue.acm.org/detail.cfm?id=971573>
 - <http://www.dcc.ufla.br/infocomp/index.php/INFOCOMP/article/view/68>
-

Obtida de "https://pt.wikipedia.org/w/index.php?title=Modelo_em_cascata&oldid=62398180"

Esta página foi editada pela última vez às 01h37min de 10 de novembro de 2021.

Este texto é disponibilizado nos termos da licença Atribuição-CompartilhaIgual 3.0 Não Adaptada (CC BY-SA 3.0) da Creative Commons; pode estar sujeito a condições adicionais. Para mais detalhes, consulte as condições de utilização.

- Política de privacidade
- Sobre a Wikipédia
- Avisos gerais
-
- Programadores
- Estatísticas
- Declaração sobre "cookies"