

DESCREVER OS MODELOS PRESCRITIVOS DE PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

ENGENHARIA DE SOFTWARE

Os modelos de Processo de Desenvolvimento de Software têm este como principal produto no contexto da Engenharia de Software.

Diferentemente de produtos de outras engenharias, o software possui alta volatilidade em função dos requisitos, visto que é constantemente impactado pelas evoluções na tecnologia, fatores estes que agregam a ele uma complexidade adicional.

A melhor tratativa para a complexidade é a aplicação de metodologia que permita a decomposição do problema em outros menores e de forma sistemática, cabendo à Engenharia de Software tal sistematização.

A Engenharia de Software é uma tecnologia em camadas, então vejamos as descrições de tais camadas:

Camada qualidade

Garante o cumprimento dos requisitos que atendem às expectativas dos usuários.

Camada de processo

Determina as etapas de desenvolvimento do software.

Camada de métodos

Define, por exemplo, as técnicas de levantamento de requisitos, os artefatos gerados em função da técnica de modelagem adotada, tais como modelos de casos de uso ou de classes.

Camada ferramentas

Estimula a utilização de ferramentas “CASE” (*Computer-Aided Software Engineering*) no desenho dos diversos artefatos ou mesmo na geração automática de código, entre outras aplicações.

Atenção

Vale destacar que a base da Engenharia de Software é a camada de processo.

PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

O processo de desenvolvimento de software é iniciado com especificações e modelos com alto nível de abstração e, à medida que o desenvolvimento de software se aproxima da codificação, o nível de abstração diminui, de modo que o código representa o nível mais baixo da abstração ou de maior detalhamento na especificação do software.

As atividades típicas que compõem o processo de desenvolvimento de software, descritas por Pressman (2016), estão representadas na Figura 1. O objetivo é ilustrar as atividades mais comuns que compõem os processos de desenvolvimento de software, ou seja, as atividades genéricas incorporadas por qualquer processo, sendo uma notória diferença entre os modelos, a ênfase dada a cada atividade.



Figura 1 – Atividades Típicas de um Processo de Desenvolvimento de Software. Fonte: Autor

Vamos agora descrever cada uma das atividades comumente previstas em um processo de desenvolvimento de software:

COMUNICAÇÃO

As primeiras atividades de um processo de software requerem uma comunicação intensiva com os usuários, buscando o entendimento do problema, a definição de objetivos para o projeto, bem como a identificação de requisitos.

PLANEJAMENTO

Destaca-se, nesta atividade, a área de conhecimento e gerenciamento de projeto, que permitirá a elaboração de um Plano de Gerenciamento do Projeto de forma sistemática, tendo como entrega importante, o cronograma que inclui as atividades a serem desenvolvidas nesse projeto.

MODELAGEM

A engenharia tem como melhor prática a geração de modelos, tal como a planta baixa de uma casa, e a maioria desses modelos gráficos, denominados de diagramas na Engenharia de Software, podendo ser complementados por descrições textuais, assim como o modelo de casos de uso.

CONSTRUÇÃO

A partir dos modelos gerados, é realizada a construção ou implementação do software, portanto, os modelos determinam o comportamento deste. Essa atividade inclui a codificação e os testes de software de acordo com o planejado.

ENTREGA

Ao final, ocorre o objetivo de um plano de projeto de software, i.e., a entrega do produto em produção de acordo com o planejado.

MODELOS DE PROCESSOS PRESCRITIVOS

A Engenharia de Software permite sistematizar o desenvolvimento de software por meio de processos. Nesse sentido, um modelo de processo identifica as atividades que serão desenvolvidas, sendo comumente utilizadas as atividades ilustradas na figura anterior.

Podemos, então, afirmar que um modelo de processo sistematiza o trabalho de engenharia de software, incluindo as atividades, fluxos de processos, artefatos gerados e outros, propiciando, também, estabilidade, controle e organização ao referido trabalho.

O que diferencia os modelos de processos? Embora todas as atividades típicas sejam utilizadas em todos os modelos, cada representação enfatiza de forma diferente cada atividade, e o encadeamento dessas ações — talvez o mais importante — é denominado de fluxo de processo.

O primeiro conjunto de modelos que abordaremos são os denominados modelos de processos prescritivos, por prescreverem metodologias, tarefas, artefatos, garantia da qualidade e mecanismos de controle de mudanças, ou seja, um conjunto de elementos de processo.

Modelo em cascata

O modelo em cascata sugere uma abordagem sequencial, isto é, a execução das atividades ilustradas na Figura 1 de forma encadeada. Este é o modelo de processo mais antigo da engenharia de software e utilizado de forma intensa à época em que o paradigma de desenvolvimento estruturado era dominante na indústria de software.

Podemos descrever algumas características negativas relativas a esse modelo, tais como:

- Projetos reais raramente seguem um fluxo sequencial, em função da volatilidade dos requisitos.
- Assume que é possível ao cliente explicitar detalhadamente todas as suas necessidades em forma de requisitos, antes do início das demais fases do desenvolvimento, aumentando a possibilidade de propagação de erros pelas fases do processo.
- Inflexibilidade na divisão do projeto em fases distintas, visto que uma versão do software somente estará disponível ao final do projeto.



Dica

Podemos aplicar o modelo em cascata em projetos com requisitos fixos e de forma que o fluxo de trabalho possa ser realizado sequencialmente até o seu encerramento.

Modelos de processo incremental e Modelos de processo evolucionário

Ambos os modelos dividem o desenvolvimento de um produto de software em ciclos, e cada ciclo de desenvolvimento inclui todas as atividades contidas na figura anterior.

Podemos observar na Figura 2 a repetição, a cada incremento e de forma iterativa, das atividades típicas de um processo de software. Cada iteração considera um subconjunto de requisitos, ocorrendo uma entrega a cada final de ciclo, isto é, a cada novo incremento um novo conjunto de funcionalidades é disponibilizado ao usuário final.

O desenvolvimento atual do software comumente tem restrições de prazos e uma alta volatilidade dos requisitos, limitando a geração de um produto abrangente. Esse modelo permite a geração de uma versão limitada que imediatamente agrega valor ao usuário.

Qual é a diferença entre o modelo incremental e evolucionário?

Modelo incremental

No modelo incremental, ocorre a entrega de um produto essencial, estando as demais iterações bem definidas.

Modelo evolucionário

No modelo evolucionário, uma parte dos requisitos está bem entendida, ocorrendo o versionamento em função desse entendimento. A cada nova iteração ocorre um melhor entendimento do problema por meio da definição de novos requisitos e novas versões são geradas.

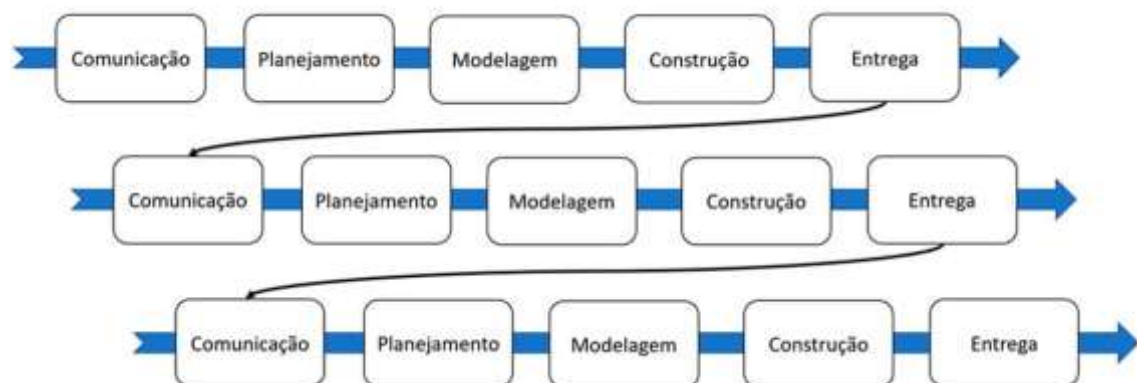


Figura 2 – O modelo incremental. Fonte: Autor

Prototipação

Imagine a construção de uma grande barragem onde, em uma fase inicial, é construído um protótipo em escala reduzida para numericamente avaliar os esforços existentes. Depois dos testes, a maquete serve como peça de museu ou vai para um laboratório de uma universidade para estudos de caso.

Você poderia questionar: como faço isso no software? A abstração é a mesma, porém o produto software tem suas peculiaridades.

Vamos imaginar que um requisito não funcional determina que a replicação de dados ocorrerá por meio de uma tecnologia desconhecida da equipe de desenvolvimento. Seria razoável aguardar o momento do desenvolvimento para realizar a tratativa desse requisito?

Com certeza não, pois existiria um risco enorme do projeto atrasar em função do atendimento desse requisito tardiamente. Nesse caso, podemos aplicar a prototipação.

Na prototipação, o desenvolvedor interage diretamente com o usuário, escutando seus pedidos e desenvolvendo, imediatamente, um protótipo do produto desejado, isso de forma iterativa e evolucionária, como um “projeto rápido” apresentado a seguir.

Saiba mais

Podemos descartar esse software gerado? Depende, normalmente essa primeira versão do protótipo serve apenas para validar um requisito e necessita ser reconstruída, mas, considerando as peculiaridades do software, essa mesma versão pode ser refinada e integrada ao sistema em desenvolvimento.

Atenção, a prototipação pode ser utilizada como modelo de processo na solução de problemas de baixa complexidade.

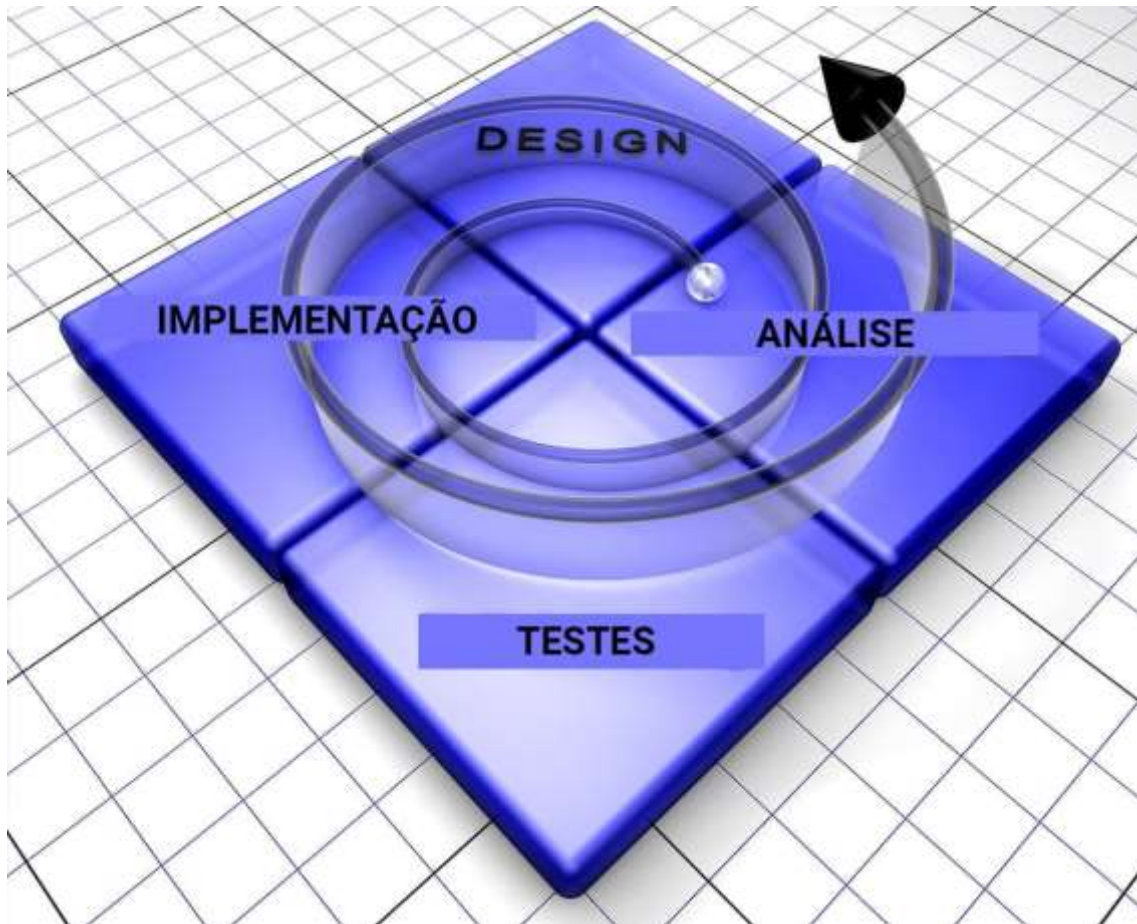


Figura 3 – O paradigma prototipação

Modelo Espiral

O Modelo Espiral foi originalmente proposto por **Barry Boehm** em 1988. De acordo com a Figura 4, cada quadrante da espiral corresponde a uma etapa do desenvolvimento e cada loop na espiral representa uma fase do processo de software. O loop mais interno pode estar relacionado com a viabilidade do sistema, o próximo, ao levantamento de requisitos e assim sucessivamente.

Primeira etapa

A primeira etapa (*Analysis*) inclui a determinação de objetivos, alternativas, riscos e restrições, em que ocorre o comprometimento dos envolvidos e o estabelecimento de uma estratégia para alcançar os objetivos.

Segunda etapa

Na segunda etapa (*Evaluation*), avaliação de alternativas, identificação e solução de riscos, executa-se uma análise de risco, sendo a prototipação uma boa ferramenta para tratar de ameaças.

Terceira etapa

Na terceira etapa (*Development*) ocorre o desenvolvimento do produto.

Quarta etapa

Na quarta etapa (*Planning*) o produto é avaliado, sendo realizado o planejamento para início de um novo ciclo.



Figura 4 – O modelo espiral. Fonte: Wikimedia Commons

Podemos fazer uso da abstração de forma a visualizarmos o modelo espiral como um metamodelo, tendo como exemplo o proposto por Pressman (2016), apresentado na Figura 5, na qual a espiral do modelo é dividida nas atividades genéricas ilustradas na Figura 1.

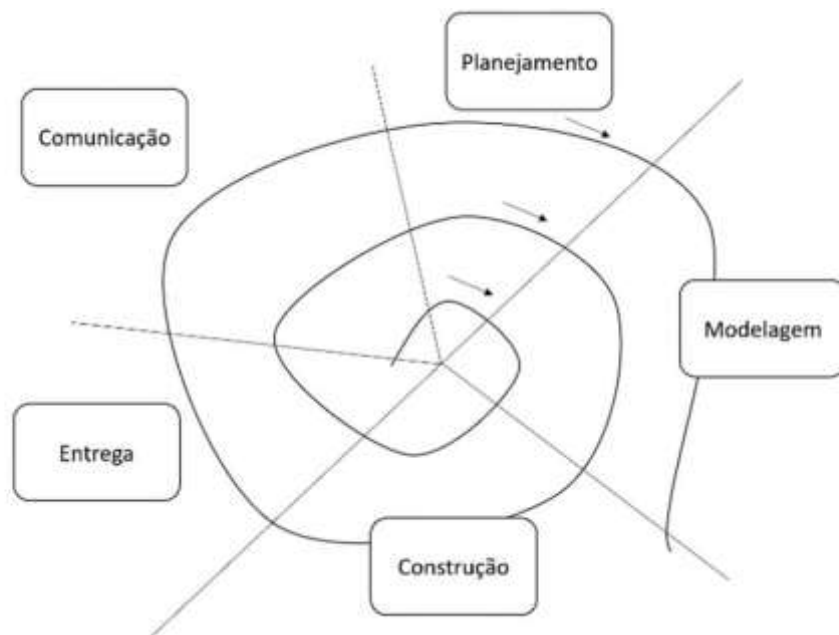


Figura 5 – O modelo espiral. Fonte: Autor

RESUMINDO

Neste módulo, pudemos destacar a importância dos modelos dos processos prescritivos para a Engenharia de Software. Vale lembrar que a camada base dessa engenharia é a referente a processos. Os modelos de processos permitem definir as atividades que serão executadas durante o projeto de software, bem como o fluxo de processos que define o encadeamento relativo a essas mesmas atividades.

Os modelos de processos prescritivos incluem a definição de metodologias, tarefas, artefatos, garantia da qualidade e mecanismos de controle de mudanças, isto é, um conjunto de elementos de processo.

O modelo em cascata determina a realização das tarefas de forma sequencial. Já os modelos de processo incremental e os modelos de processo evolucionário permitem um versionamento do software de forma iterativa e incremental.

O paradigma da prototipação é aplicado primariamente na validação de requisitos. O modelo espiral permite um desenvolvimento evolucionário do software.