**UNIVERSIDADE DO ESTADO DE MINAS GERAIS-UEMG**

**NÚCLEO ACADÊMICO DE TECNOLOGIA E ENGENHARIA**

**CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**Modelos de Processos de Desenvolvimento de Software**

***PROTOTIPAÇÃO***

**Alunos**

Anderson Veloso dos Santos

Junior César da Silva

Maria Andressa de Paula Silva

Rafael de Oliveira Romano

**PASSOS/MG**

**2016**

**1 Modelos de Processos de Desenvolvimento de Software**

A engenharia de *software* tem como base a camada de processos, mas também engloba métodos de gerenciamento e desenvolvimento de *software*, bem como ferramentas (PRESSMAN, 2011).

O processo está para a metodologia de desenvolvimento do produto, mas não obrigatoriamente a como desenvolver o *software*. Os processos da engenharia de *software* possibilitam à equipe desenvolvedora criar suas próprias tarefas e ações, garantindo assim um desenvolvimento de alta qualidade dentro do prazo estipulado. Os métodos envolvem tarefas de comunicação, planejamento, modelagem, construção e entrega. Já as ferramentas dão suporte aos processos e métodos.

Segundo Sommerville (2011), para qualquer escolha de modelo de processo de *software* escolhido em um projeto, este se baseará nas seguintes etapas:

* Especificação: requisitos e funcionalidades;
* Projeto: como o *software* deverá ser desenvolvido;
* Implementação: codificação do *software*;
* Validação: validação frente aos requisitos;
* Evolução: visa manter/corrigir e evoluir o *software*.

**2 Modelos de Processos Gerais**

Um modelo de processo de *software* representa as atividades que serão utilizadas no desenvolvimento e podem ser adaptados de acordo com a necessidade. Os modelos podem ser classificados em:

* Artesanais: não há aplicação da Engenharia de Software, também conhecido como modelo “gambiarra”.
* Tradicionais: maioria dos conceitos e práticas da Engenharia de Software, estes criados a partir do ano de 1970.
* Ágeis: modelos modernos, novas práticas e abordagens. Criado a partir de opiniões de Engenheiros de Software e grupos de desenvolvedores.

Os modelos mais usados são: *i)* Modelo Cascata (baseado em atividades sequenciais), *ii)* Desenvolvimento Incremental (baseado em atividades intercaladas), *iii)* Orientado a Reuso (baseado na existência de componentes reusáveis). Ainda assim, os modelos gerais não são de uso exclusivos, dando a opção de combinação entre diferentes modelos para as organizações durante o processo de desenvolvimento.

**3 Prototipação**

O cliente define uma série de objetivos gerais para o *software*, mas não é capaz de detalhar os requisitos para funções e recursos. Este é um dos casos em que a prototipação poderá ser abordada.

Um protótipo é uma versão inicial de um *software* que permite a detecção precoce de problemas, reduzindo custos e melhorando a qualidade do produto. O desenvolvimento rápido e iterativo do protótipo garante aos *stakeholders* do sistema experimentá-lo no início do processo. O protótipo pode ajudar em duas ocasiões: *i)* no processo de engenharia de requisitos, pode ajudar na elicitação e validação de requisitos. *ii)* no processo de projeto de sistema, usado para estudar soluções específicas do *software* e para apoiar o projeto de interface de usuário.

A prototipação começa com a comunicação. O engenheiro de *software* e o cliente encontram-se e definem os objetivos gerais do *software*, identificam as necessidades conhecidas e delineiam áreas que necessitam de mais definições. Uma iteração de prototipação é planejada rapidamente e a modelagem (na forma de um projeto rápido) ocorre. O projeto rápido concentra-se na representação daqueles aspectos do *software* que estarão visíveis para o cliente/usuário (por exemplo, *layout* da interface humana ou formatos de saída de tela). O projeto rápido leva a construção de um protótipo, que é implantado e depois avaliado pelo cliente/usuário. O *feedback* é usado para refinar os requisitos do *software*. A iteração ocorre à medida que o protótipo é ajustado para satisfazer ás necessidades do cliente, e, ao mesmo tempo permite ao desenvolvedor entender melhor o que precisa ser feito.

Estabelecer objetivos do protótipo

Desenvolver protótipo

Definir funcionalidades do protótipo

Avaliar o protótipo

Processo de desenvolvimento de protótipo

Idealmente, o protótipo serve como um mecanismo para identificação dos requisitos do *software*. Se um protótipo executável é elaborado, o desenvolvedor tenta usar partes de programas existentes ou aplicar ferramentas (por exemplo, geradores de relatórios, gestores de janelas etc.) que possibilitem programas executáveis serem gerados rapidamente.

O protótipo pode servir como “o primeiro sistema”, e recomenda-se que este seja descartado. Mas essa pode ser uma visão idealizada. É verdade que tanto cliente quanto desenvolvedor, gostam do paradigma de prototipação. Os usuários têm o sabor de um sistema real e os desenvolvedores conseguem construir algo imediatamente. Ainda assim, a prototipação pode ser problemática pelas seguintes razões:

* O cliente vê o que parece ser uma versão executável do *software*, ignorando que o protótipo apenas consiga funcionar precariamente, sem saber que na pressa de fazê-lo rodar, ninguém considerou a sua qualidade global ou sua facilidade de manutenção no longo prazo. Quando informamos de que o produto deve ser refeito para que altos níveis de qualidade possam ser atingidos, o cliente reclama e exige “alguns consertos”, para transformar o protótipo num produto executável. Em geral, a gerência de desenvolvimento de *software* concorda.
* O desenvolvedor frequentemente faz concessões na implementação a fim de conseguir rapidamente um protótipo executável. Um sistema operacional, ou uma linguagem de programação inapropriada, pode ser usado simplesmente por estar disponível e ser conhecido; um algoritmo ineficiente pode ser implementado simplesmente para demonstrar capacidade. Passados certo tempo, o desenvolvedor pode ficar familiarizado com essas escolhas e esquecer todas as razões porque eram inadequadas. As escolhas passam a ser insuficientes e se tornam parte integral do sistema.

Apesar de problemas poderem ocorrer, a prototipação pode ser um paradigma efetivo para a Engenharia de Software. O importante é definir as regras do jogo no início; isto é, cliente e desenvolvedor devem estar de acordo que o protótipo é construído para servir como mecanismo de definição dos requisitos. Depois ele será descartado, e o *software* real será submetido à engenharia com um olho na qualidade.

**3.1 Vantagens e Desvantagens da Prototipação**

O uso de prototipação como modelo de desenvolvimento de *software* pode trazer as seguintes vantagens:

* Melhora a qualidade da especificação do *software* a ser desenvolvido, contribuindo para uma queda nos custos de desenvolvimento e manutenção;
* Antecipa o treinamento dos usuários;
* Custos mais baixos;
* Funcionalidade completa;
* Facilita uma resposta mais rápida dos desenvolvedores;
* Partes do protótipo podem ser aproveitadas no desenvolvimento do sistema.

Em contrapartida, as desvantagens observadas são:

* O custo na maioria dos casos é considerado alto;
* Verificação de erros limitada;
* Verificação de código fraca;
* O cliente tende a confundir o protótipo com uma versão do sistema, isso cria falsas expectativas com relação a prazos;
* Equipe de desenvolvedores e usuários podem perder entusiasmo após a apresentação de várias versões de protótipos;
* Pode haver uma redução da disciplina da equipe que tende a enxergar a prototipação como um “treino” que não é para valer.

**3.2 Quando utilizar a prototipação**

Apesar de a prototipação poder ser usada como modelo de processo independente, ela é mais comumente usada como uma técnica que pode ser implantada dentro do contexto de qualquer um dos modelos de processo de *software*. Independentemente da maneira como é aplicado, a prototipação auxilia o engenheiro de *software* e o cliente a entenderem melhor o que deve ser construído quando os requisitos estão confusos.

Para fazer um produto que realmente atenda a todas as necessidades do cliente, a prototipação poderá ser abordada quando o desenvolvedor não tem certeza quanto à eficiência de um algoritmo, ou quanto à adaptabilidade de um sistema operacional, ou ainda quanto à forma em que deva ocorrer a interação entre o cliente e o sistema.

Deverá também ser abordada quando o cliente não for capaz de gerar requisitos definidos, de entrada, processamento e saída, para o sistema (*software*); ou durante a comunicação com alguma interface, periféricos/componentes. Interação homem-máquina pode não ser aceita pelo cliente, ou seja, a interface de comunicação com a aplicação (s*oftware*) pode ser confusa ou não usual.

**3.3 Quando o uso de prototipação não é recomendado**

De forma geral, o protótipo auxilia na identificação dos requisitos do *software*. Os protótipos podem ser descartados quando os usamos apenas para entender um determinado requisito ou pode ser utilizado como um produto evolucionário que servirá para o cliente.

**4 Conclusão**

A Engenharia de Software presta suporte aos engenheiros e desenvolvedores através dos modelos de processos de desenvolvimento de *software*. Estes modelos, também chamados de ciclo de vida de um *software*, incluem processos, métodos e ferramentas. A escolha adequada de qual modelo usar vai depender do tipo de projeto, da equipe desenvolvedora e também do próprio cliente, este que muitas vezes não consegue contribuir para o projeto somente através dos requisitos.

A fim de atender todas as necessidades do cliente, o modelo de prototipação vem auxiliar o engenheiro de *software* e o cliente a entenderem melhor o que deve ser construído quando os requisitos estão confusos. Este modelo poderá ser usado independentemente ou usado como uma técnica a ser implantada junto a qualquer outro modelo de processo de *software*.

Um protótipo é uma versão inicial de um *software* que permite a detecção precoce de problemas, reduzindo custos e melhorando a qualidade do produto. Seu desenvolvimento se baseia em um plano de prototipação (comunicação entre o engenheiro de software e o cliente), definição geral (funcionalidade do protótipo), protótipo executável (desenvolvimento) e relatório de avaliação (avaliação do protótipo).

De modo geral, seu uso poderá ser aderido quando o cliente não for capaz de gerar os requisitos definidos, quando o desenvolvedor estiver inseguro quanto a eficiência de um algoritmo, quanto à adaptabilidade de um sistema operacional, ou ainda quanto à forma em que deva ocorrer a interação entre o cliente e o sistema. Poderá ser descartado quando seu uso for utilizado apenas para entender um determinado requisito.

**Referências**

FIGUEIREDO, Eduardo. **Processos de Software**. Disponível em: < http://homepages.dcc.ufmg.br/~figueiredo/disciplinas/aulas/processos-software\_v01.pdf>. Acessado em 26 de Março. 2016.

GARCIA, Luís F. F. **Engenharia de Software I**. Canoas, ULBRA, 2013.

PÁDUA, Clarindo I. P. S. **Prototipação**. Disponível em: <http://homepages.dcc.ufmg.br/~clarindo/arquivos/disciplinas/eu/material/transparencias/topicos/9-prototipacao.pdf>. Acessado em 25 de Março. 2016.

PRESSMAN, Roger S. **Engenharia de Software**. 7ª edição. São Paulo, Bookman, 2011.

SOMMERVILLE, Ian. **Engenharia de Software**. 9ª edição. São Paulo, Pearson, 2011.