**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.

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.

**2 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.

O paradigma da prototipagem 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 prototipagem 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.

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 clientes quanto desenvolvedores gostam do paradigma de prototipagem. Os usuários têm o sabor de um sistema real e os desenvolvedores conseguem construir algo imediatamente.

Apesar de problemas poderem ocorrer, a prototipagem 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 (pelo menos em parte), e o software real será submetido à engenharia com um olho na qualidade.

**2.1 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.

**2.2 Quando utilizar a prototipação**

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 (Software) pode ser confusa ou não usual.

**2.3 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.

**Referencias**

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