A importância da Engenharia de Software na Computação nos dias atuais

Rafael Barasuol Rohden

Universidade Regional do Noroeste do Estado do Rio Grande do Sul (UNIJUI)
CEP 98700-000 – Ijuí – RS – Brasil

Departamento de Tecnologia – DETEC

rafaelrohden@gmail.com

Resumo. *Este pequeno artigo descreve sucintamente a importância da engenharia de software nos tempos atuais onde o desenvolvimento de software se destaca na sociedade contemporânea. Destaca a preocupação com a qualidade do produto "software". Descreve os processos e modelos de processos no desenvolvimento de software para atingir o objetivo de qualidade de software.*

Abstract. *This little paper briefly describes the importance of Software Engineering on this time where the development of software is well-known and used over the world. Highlights the concern with the quality of the product software. Describes the processes and process models in software development to reach the goal of software quality.*

1. Introdução

O desenvolvimento de software tem crescido nos últimos anos devido a sua grande importância na sociedade contemporânea. O uso cada vez maior de computadores pessoais e nas diversas áreas do conhecimento humano tem gerado uma crescente demanda por soluções que automatizem os diversos processos.

Iniciantes da área de desenvolvimento de software têm o costume de confundir desenvolvimento com programação, pois estes estão em fase de desenvolver suas habilidades no raciocínio lógico na resolução de pequenos problemas. Mas quando se deparam com problemas mais complexo requerem maior conhecimento e habilidades já que o uso de uma abordagem individual, centrada na programação não é mais indicada.

Segundo Falbo (2005), com o intuito de melhorar a qualidade dos softwares em geral e aumentar a produtividade no desenvolvimento de tais produtos, surgiu a *engenharia de software*. É responsável estabelecimento de técnicas e práticas para o desenvolvimento de software cobrindo uma ampla área de aplicações e diferentes tipos de dispositivos, tais como sistemas de informação corporativos, sistemas e portais Web, aplicações em telefones celulares.

2. Engenharia de Software

A engenharia de software propõe métodos sistemáticos com o uso adequado de ferramentas e técnicas, que levam em consideração o problema a ser resolvido, as necessidades dos clientes e os recursos disponíveis.

A área está fundamentada, sobretudo na ciência da computação e na matemática. Ao longo dos últimos anos, essa área e suas diferentes disciplinas têm amadurecido bastante, através da proposição de novos métodos e técnicas que possibilitem o desenvolvimento de softwares mais confiáveis, de melhor qualidade, com custo reduzido e alta produtividade.

Qualidade é o que motiva os desenvolvedores usarem dos processos descritos pela Engenharia de Software. Todos os processos descritos nas diferentes técnicas e formas de desenvolvimentos buscam o mesmo objetivo, a qualidade do software, qualidade que vai desde a produção até a entrega do produto (software). É necessário, segundo Falbo (2005), que a qualidade seja incorporada ao produto ao longo de seu processo de desenvolvimento. E que, de fato, a qualidade dos produtos de software depende fortemente da qualidade dos processos usados para desenvolvê-los e mantê-los.

Segundo Falbo (2005), um processo de software pode ser classificado quanto ao seu propósito em:

- Atividades de Desenvolvimento (ou Técnicas ou de Construção): são as atividades diretamente relacionadas ao processo de desenvolvimento do software, ou seja, que contribuem diretamente para o desenvolvimento do produto de software a ser entregue ao cliente. São exemplos de atividades de desenvolvimento: especificação e análise de requisitos, projeto e implementação.
- Atividades de Gerência: são aquelas relacionadas ao planejamento e acompanhamento gerencial do projeto, tais como realização de estimativas, elaboração de cronogramas, análise dos riscos do projeto etc.
- Atividades de Garantia da Qualidade: são aquelas relacionadas com a garantia da qualidade do produto em desenvolvimento e do processo de software utilizado, tais como revisões e inspeções de produtos (intermediários ou finais) do desenvolvimento.

3. Processos de Software

Para manter a qualidade da produção de um software e garantir também um software de qualidade é preciso seguir algumas normas, algumas regras comum a todos aqueles que querem ter seu software com certificado de qualidade.

O processo de software é definido, por Macoratti.net, como um conjunto de atividades uniformizadas a serem aplicadas sistematicamente que se encontram agrupadas em fases, cada uma das quais com os seus intervenientes com responsabilidades, que possui diversas entradas e produz diversas saídas. Isto é, define quem faz o quê, quando e como para atingir certo objetivo.

- Projeto de Interface: onde cada módulo tem sua interface de comunicação estudada e definida.
- Projeto Detalhado: onde os módulos em si são definidos, e possivelmente traduzidos para pseudo-código.

- Implementação
  - Codificação: a implementação em si do sistema em uma linguagem de computador.

- Validação
  - Teste de Unidade e Módulo: a realização de testes para verificar a presença de erros e comportamento adequado a nível das funções e módulos básicos do sistema.
  - Integração: a reunião dos diferentes módulos em um produto de software homogêneo, e a verificação da interação entre estes quando operando em conjunto.

- Manutenção e Evolução
  - Nesta fase, o software em geral entra em um ciclo iterativo que abrange todas as fases anteriores.

Desta forma as atividades relacionadas a um processo de software estão diretamente vinculadas com a produção do software como produto final. Mantendo o propósito principal que é a qualidade. Traveses da organização destas atividades concentram-se energias das pessoas envolvidas em atividades especificas sempre focando as pessoas na qualidade.

3.1.2 Modelos de Processo de Desenvolvimento de Software

Os modelos de processos de desenvolvimento de software surgiram pela necessidade de dar resposta às situações a analisar, porque só na altura em que enfrentamos o problema é que podemos escolher o modelo.

- Modelo cascata: tem como principal característica a seqüência de atividades onde cada fase transcorre completamente e seus produtos são vistos como entrada para uma nova fase.
- Modelo espiral: série de pequenos ciclos, cada um finalizando um versão de um software executável. O modelo em espiral assume que o processo de desenvolvimento ocorre em ciclos, cada um contendo fases de avaliação e planejamento, onde a opção de abordagem para a próxima fase (ou ciclo) é determinada.
- Modelo de do processo de desenvolvimento iterativo e incremental: é mais prático dividir o trabalho em partes menores ou iterações. Cada iteração resultará num incremento. O princípio subjacente ao processo incremental e iterativo é que a equipa envolvida possa refinar e alargar paulatinamente a qualidade, detalhe e âmbito do sistema envolvido.

- Projeto de Interface: onde cada módulo tem sua interface de comunicação estudada e definida.
- Projeto Detalhado: onde os módulos em si são definidos, e possivelmente traduzidos para pseudo-código.

- Implementação
  - Codificação: a implementação em si do sistema em uma linguagem de computador.

- Validação
  - Teste de Unidade e Módulo: a realização de testes para verificar a presença de erros e comportamento adequado a nível das funções e módulos básicos do sistema.
  - Integração: a reunião dos diferentes módulos em um produto de software homogêneo, e a verificação da interação entre estes quando operando em conjunto.

- Manutenção e Evolução
  - Nesta fase, o software em geral entra em um ciclo iterativo que abrange todas as fases anteriores.

Desta forma as atividades relacionadas a um processo de software estão diretamente vinculadas com a produção do software como produto final. Mantendo o propósito principal que é a qualidade. Traveses da organização destas atividades concentram-se energias das pessoas envolvidas em atividades especificas sempre focando as pessoas na qualidade.

3.1.2 Modelos de Processo de Desenvolvimento de Software

Os modelos de processos de desenvolvimento de software surgiram pela necessidade de dar resposta às situações a analisar, porque só na altura em que enfrentamos o problema é que podemos escolher o modelo.

- Modelo cascata: tem como principal característica a seqüência de atividades onde cada fase transcorre completamente e seus produtos são vistos como entrada para uma nova fase.
- Modelo espiral: série de pequenos ciclos, cada um finalizando um versão de um software executável. O modelo em espiral assume que o processo de desenvolvimento ocorre em ciclos, cada um contendo fases de avaliação e planejamento, onde a opção de abordagem para a próxima fase (ou ciclo) é determinada.
- Modelo de do processo de desenvolvimento iterativo e incremental: é mais prático dividir o trabalho em partes menores ou iterações. Cada iteração resultará num incremento. O princípio subjacente ao processo incremental e iterativo é que a equipa envolvida possa refinar e alargar paulatinamente a qualidade, detalhe e âmbito do sistema envolvido.

- Modelo de Prototipagem: idéia básica deste modelo é que ao invés de manter inalterados os requisitos durante o projeto e codificação, um protótipo é desenvolvido para ajudar no entendimento dos requisitos. Este desenvolvimento passa por um projeto , codificação e teste, sendo que cada uma destas fases não é executada formalmente.

4. Estimativas

Antes mesmo de serem iniciadas as atividades técnicas de um projeto, o gerente e a equipe de desenvolvimento devem estimar o trabalho a ser realizado, os recursos necessários, o tempo de duração e, por fim, o custo do projeto. Apesar das estimativas serem um pouco de arte e um pouco de ciência, esta importante atividade não deve ser conduzida desordenadamente. As estimativas podem ser consideradas a fundação para todas as outras atividades de planejamento de projeto

Para alcançar boas estimativas de prazo, esforço e custo, existem algumas opções, segundo Falbo (2005):

1. Postergar as estimativas até o mais tarde possível no projeto.
2. Usar técnicas de decomposição.
3. Usar um ou mais modelos empíricos para estimativas de custo e esforço.
4. Basear as estimativas em projetos similares que já tenham sido concluídos.

Quando se fala em estimativas, está-se tratando na realidade de diversos tipos de estimativas: tamanho, esforço, recursos, tempo e custos. Geralmente, a realização de estimativas começa pelas estimativas de tamanho. A partir delas, estima-se o esforço necessário e, na seqüência, alocam-se os recursos necessários, elabora-se o cronograma do projeto (estimativas de tempo) e, por fim, estima-se o custo do projeto.

5. Implantação e testes

É muito importante, segundo Falbo (2005) que haja padrões organizacionais para a fase de implementação. Esses padrões devem ser seguidos por todos os programadores e devem estabelecer, dentre outros, padrões de nomes de variáveis, formato de cabeçalhos de programas e formato de comentários, recuos e espaçamento, de modo que o código e a documentação a ele associada sejam claros para quaisquer membros da organização.

Ainda que padrões sejam muito importantes, deve-se ressaltar que a correspondência entre os componentes do projeto e o código é fundamental, caracterizando-se como a mais importante questão a ser tratada. O projeto é o guia para a implementação, ainda que o programador deva ter certa flexibilidade para implementá-lo como código

Após a implantação é hora dos testes, teste é uma atividade de verificação e validação do software e consiste na análise dinâmica do mesmo, isto é, na execução do produto de software com o objetivo de verificar a presença de defeitos no produto e aumentar a confiança de que o mesmo está correto

Porem, mesmo se um teste não detectar defeitos, isso não quer dizer necessariamente que o produto é um produto de boa qualidade. Muitas vezes, a

atividade de teste empregada pode ter sido conduzida sem planejamento, sem critérios e sem uma sistemática bem definida, sendo, portanto, os testes de baixa qualidade

6. Conclusão

A engenharia de software é a disciplina do conhecimento humano que tem por objetivo definir e exercitar processos (humanos atuando como máquinas), métodos (planos de processos), ferramentas e ambientes (máquinas apoiando processos e métodos) para construção de software que satisfaça necessidades de cliente e usuário dentro de prazos e custos previsíveis.

Através de processos de software, que descrevem atividades padronizadas sendo estas aplicadas sistematicamente e agrupadas em fases, cujas quais possuem diferentes entradas e diversas saídas como solução, juntamente com pessoas altamente capacitadas as equipes de desenvolvimento conseguem manter uma qualidade do software.

É imprescindível o desenvolvimento de software nos dias atuais sem o uso da engenharia de software. A concorrência devido a globalização, e o crescente do numero de usuários de computadores trazem aos desenvolvedores uma árdua tarefa de criar aplicativos (soluções) com qualidade e adequado as necessidades dos usuários.

Referências Bibliográficas

Falbo, Ricardo de Almeida. (2005) "Engenharia de Software - Notas de Aula", Universidade Federal do Espírito Santo, http://www.inf.ufes.br/~falbo/download/aulas/es-g/2005-1/NotasDeAula.pdf. Acessado em: 04/10/2009

PRESSMAN, R. S. , Software engineering: A practitioner's approach. 4th. ed. McGraw-Hill, 1997. p. 22-53.

SCHWARTZ, J. I. ,Construction of software. In: *Practical Strategies for Developing Large Systems*. Menlo Park: Addison-Wesley, 1st. ed., 1975.

Macoratti.net "O processo de Software", _____, Acessado em: 04/10/2009.