|  |
| --- |
| USJT - Faculdade de Tecnologia e Ciências Exatas  Curso - Especialização em Engenharia de Software  Disciplina - Métricas de Tamanho, Complexidade e Qualidade de Software.  Prof. Edson Saraiva de Almeida |
| **Tema:** Plano de Métricas  Objetivos de aprendizagem: Compreender a importância da comunicação aberta e contínua com dados objetivos diretamente das atividades em progresso e das configurações do produto em desenvolvimento.   1. **Gerenciamento de projetos**   Segundo Sommerville (2007) a natureza do software torna o gerenciamento de projetos de software mais difícil do que o gerenciamento de projetos de outros tipos de engenharia. Essas distinções tornam o gerenciamento de software particularmente difícil. Algumas direfenças são: o produto é intangível, não existe um processo padrão de software, projetos de software de grande porte são frequentemente projetos únicos.   1. **Mascos e produtos a serem entregues**   Os gerentes precisam de informações para realizar seu trabalho. Como o software é intangível, essas informações podem ser fornecidas apenas como relatórios e documentos que descrevem o estado do software que é desenvolvido. Sem essas informações é impossível avaliar se o trabalho esta progredindo de acordo com as estimativas de custos e cronograma. O planejamento do projeto envolve estabelercer uma série de marcos. Uma marco é um ponto final reconhecível de uma atividade do processo de desenvolvimento do software. A cada marco, deve escistir uma saída formal, como um relatório, que possa ser apresentado a gerência. Os marcos devem representar o fim de um estágio lógico e distinto do prjeto. Para estabelecer os marcos, o processo de software deve ser decomposto em atividades básicas com saídas associadas (figura3).  [Figura 3]  O modelo IDEF0 (MARCA, 2006) pode ser usado para descrever as atividades básicas e tarefas de um processo de desenvolvimento de software.  [Figura 2] |

|  |
| --- |
| Os componentes da sintaxe do IDEF0 podem ser adaptados para descrever o processo:   * Entrada - a lista de critérios de entrada que devem estar satisfeitos antes de iniciar a atividade. * Atividade do processo - um conjunto de tarefas que devem estar satisfeitos antes de iniciar a atividade. * Saída - uma lista de critérios que devem ser atendidos para que a atividade possa ser considerada concluída. * Mecanismos - definem os recursos que são utilizados para executar a atividade do processo, podem ser humanos ou materiais. * Controle - um conjunto de medições podem ser definidos para os critérios de entrada e conclusão da tarefa o que favorece a utilização de técnicas para definições de métricas baseadas em obejtivos como GQM.   O diagrama IDEF0 permite uma decomposição das atividades do processo que compõem o diagrama.  [Figura 3]  A figura 4 descreve o modelo de processo de desenvolvimento de software Walterfall (SOMMERVILLE, 2007). Um exemplo parcial da aplicação da técnica de modelagem IDEF0 para o modelo Walterfall de desenvolvimento de software é mostrado na figura 5.  [Figura 4] |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [Figura 5]  Tabela 1 - Texto complementar para detalhar o diagrama IDEF0 - Atividade 1: Definição de requisitos   |  | | --- | | Entrada |  |  |  | | --- | --- | | Necessidades do cliente | O documento de visão descreve a visão que os envolvidos têm do produto a ser desenvolvido, em termos das necessidades e características mais importantes. (Krutchen, 2004) |  |  | | --- | | Controle |  |  |  | | --- | --- | | Padrão de documentação de casos de usos | Os casos de uso seguem o padrao para documentação de casos de uso utilizado na organização (guia de specificação de casos de usos V1.doc) |  |  | | --- | | Saídas |  |  |  | | --- | --- | | Especificação de casos de uso | Descrição das funcionalidades do sistem classificados por prioridade |  |  | | --- | | Recursos |  |  |  | | --- | --- | | Analista de requisitos | Recursos humanos envolvidos no projeto especialista no levantamento de requisitos de software |  1. Selecione um modelo de processo e descreva os marcos e produtos a serem entregues utilizando a técnica IDEF0. 2. Estabeleça um conjunto adequado de medidas, orientado pelos objeticos de medição. Utilize o padrão GQM para selecionar as métricas de acordo com o processo de desenvolvimento de software estabelecido.   Referencias bibliográficas  MARCA, David A., MCGOWAN, Clement L. IDEF0 and SADT: a modeler’s guide. OpenProcess, Incorporated, 2006.  SOMMERVILLE, Ian. Engenharia de Software-8 Edição. São Paulo: Person Addison-Wesley. 2007 |
|  |

**Requisitos => Projeto:**

- Inspeção da ata de reunião

- e/ou confrontação da ata de reunião com o mapa mental

**Projeto => Codificação:**

- Inspeção do wireframe

**Codificação => Testes**

- Classificação dos bugs nos testes

**Testes => Homologação**

Classificação dos retornos de “não conforme” do cliente

**Homologação => Operação (Produção)**

Classificação dos bugs de produção

-----

**Universidade São Judas Tadeu – Pós-Graduação em Engenharia de Software**

**Tema:** Plano de métricas

1) Selecione um modelo de processo e descreva os marcos e produtos a serem entregues utilizando a técnica IDEF0

2)Estabeleça um conjunto adequado de medidas, orientado pelos objetivos de medição. Utilize o padrão GQM para selecionar as métricas de acordo com o processo de desenvolvimento de software estabelecido

**Nome do aluno:** André de Paula Terceiro

**Nome do aluno:** Pedro A. Saraiva Jr.

**Resolução do item Requisitos => Projeto**

**Especificação dos indicadores**

**1. Objetivo da medição**

|  |  |
| --- | --- |
| Analisar | Número de defeitos |
| Com o propósito de | Contabilização de defeitos |
| Com relação | Quantidade média de defeitos por página na ata de reunião |
| Do ponto de vista | QA |
| No contexto do | Redator da ata de reunião |
| Objetivo estratégico | Diminuição de retrabalho por problemas em um documento base para o desenvolvimento do software |

**2. Especificação das medidas básicas**

**2.1 Medida QEFA**

**2.1.1 Descrição da medida**

|  |  |
| --- | --- |
| Identificação da medida | QEFA |
| Descrição | Quantidade média de erros por página de texto corrido\* na ata de reunião  \* As duas primeiras páginas da ata de reunião são uma capa e um histórico de revisões e não contam no cálculo, apesar de serem inspecionadas |
| Unidade de medida | unitário |
| Escala | números inteiros não negativos |

**2.1.2 Procedimento de coleta e armazenamento**

|  |  |
| --- | --- |
| 1 | O inspetor, membro da equipe de QA, obtem recebe um e-mail de um membro da equipe de requisitos indicando o caminho no repositório em que a ata de reunião está disponível para inspeção |
| 2 | O inspetor avalia a ata de reunião conforme o checklist de inspeção armazenado em svn://….. |
| 3 | O inspetor preenche a planilha “Registro de inspeções” do arquivo armazenado em svn://…. , registrando a inspeção |
| 4 | O inspetor preenche a planilha “Defeitos detectados” do mesmo arquivo do passo anterior, detalhando os defeitos encontrados e inserindo suas classificações:  a) Estrutural: defeito relacionado à estrutura do documento  b) Ambiguidade: informação ambigua no documento  c) Omissão: uma informação relevante para a compreensão de um trecho do documento foi omitida e não há referências de onde obtê-la  d) Inconsistência: contradição entre trechos do documento  e) Fato incorreto: algum fato declarado no documento comprovadamente não pode ser verdadeiro  f) Outros: Outro tipo de defeito que não se enquadra nas demais categorias  OBS: As instruções de preenchimento da planilha também estão no checklist indicado no passo 2 |

**2.1.3 Procedimento de análise**

|  |  |
| --- | --- |
| 1 | O inspetor extrai da planilha “Registro de inspeções” o número médio de erros por página de texto corrido, que é cálculado automaticamente após preencher os defeitos encontrados na inspeção. Sendo este número maior que o definido como limite no checklist, será necessária uma nova inspeção após as correções pelo membro da equipe de requisitos. |
| 2 | O membro da equipe de QA avisa por e-mail o membro da equipe de requisitos que a revisão está finalizada, que as correções necessárias estão disponíveis na planilha e se será necessária ou não uma nova revisão após as correções dos erros |

**2.2 Medida DTRH**

**2.2.1 Descrição da medida**

|  |  |
| --- | --- |
| Identificação da medida | DTRH |
| Descrição | Comparação dos defeitos de cada tipo por página na ata de reunião em relação ao histórico |
| Unidade de medida | percentual |
| Escala | números inteiros não negativos |

**2.2.2 Procedimento de coleta e armazenamento**

Mesmo processo de coleta e armazenamento da medida QEFA

**2.2.3 Procedimento de análise**

|  |  |
| --- | --- |
| 1 | O inspetor extrai da planilha “Gráficos” do arquivo armazenado em svn://…. , para cada tipo de defeito, a comparação entre o valor médio por página da ata analisada em relação ao histórico. Sendo indicado que houve um aumento percentual maior que 30%, o inspetor deve alertar ao redator da ata para atentar-se a este tipo de defeito |

**Resolução do item Codificação => Testes**

**Especificação dos indicadores**

**1. Objetivo da medição**

|  |  |
| --- | --- |
| Analisar | Complexidade ciclomática |
| Com o propósito de | Verificar se o código fonte apresenta indícios de baixa manutenibilidade |
| Com relação | Análise de complexidade ciclomática |
| Do ponto de vista | QA |
| No contexto do | Projeto |
| Objetivo estratégico | Maximizar a manutenibilidade |

**2.1 Medida CC**

**2.1.1 Descrição da medida**

|  |  |
| --- | --- |
| Identificação da medida | **CC** |
| Descrição | Complexidade ciclomática |
| Unidade de medida | unitário |
| Escala | números racionais |

**2.1.2 Procedimento de coleta e armazenamento**

|  |  |
| --- | --- |
| 1 | Nos dias 10 e 25 de cada mês, um membro da equipe de QA acessa a ferramenta SonarQube no endereço http://xxxxx e efetua logon |
| 2 | Este membro da equipe de QA acessa o link “xxxx” e verifica a evolução da complexidade ciclomática nos últimos 30 dias para cada projeto que atualmente esteja em desenvolvimento, conforme ilusta a imagem abaixo: |
| 3 | O membro da equipe de QA anota os valores de complexidade ciclomática média por arquivo em cada projeto e a evolução desta. |
| 4 | Este membro da equipe de QA efetua logout da ferramenta. |

**2.1.3 Procedimento de análise**

|  |  |
| --- | --- |
| 1 | O membro da equipe de QA, para cada projeto em desenvolvimento, verfica se a complexidade ciclomática por arquivo é maior que 10 atualmente e se ela não diminuiu nos últimos 30 dias. Sendo as duas condições verdadeiras, para cada projeto em que isto ocorreu, cria uma linha na planilha disponível em svn://x.x.x.x/yyyyy indicando um problema de complexidade ciclomática. |
| 2 | O membro da equipe de QA avisa envia para o e-mail [desenvolvimento@xxxxx.com](mailto:desenvolvimento@xxxxx.com) a planilha que gerou e comenta no corpo os resultados encontrados |