

Estudo de Caso de Aplicação da Métrica de Pontos de Casos de Uso numa Empresa de Software

Viviane Heimberg (Senior Sistemas)

viviane@senior.com.br

Everaldo Artur Grahl (FURB/DSC)

egrahl@furb.br

Resumo: A estimativa de tamanho é uma das métricas de software mais utilizadas, porque a partir dessa dimensão é possível definir o esforço, o prazo e os custos necessários para o desenvolvimento do software. Atualmente uma das métricas que está sendo testada e adotada nas empresas é a PCU (Pontos de Casos de Uso). Neste artigo foi apresentado um estudo de caso de uma empresa de software onde foi utilizada a métrica de Pontos de Casos de uso para aumentar a precisão nas estimativas de tempos de desenvolvimento de três projetos de software Web. O experimento demonstrou a necessidade de se realizar controles mais eficientes na obtenção e análise dos requisitos e em especial dos Casos de Uso, além de ser importante o ajuste de valores de horas para futuras previsões.

Palavras-chave: Métricas; Pontos de casos de uso.

1 Introdução

Na última década, a mudança na tecnologia de comunicação com a popularização da internet influenciou profundamente os processos nas empresas exigindo novas metodologias de análise e desenvolvimento de sistemas de informação. As empresas precisam adequar seus sistemas atuais aos novos volumes de informação, tanto estratégicas quanto operacionais. Segundo Tavares, Carvalho e Castro (2004), este pode ser considerado um desafio para as organizações que desenvolvem softwares, pois precisam construir sistemas úteis e no tempo adequado para atender os níveis de competitividade exigidos pelo mercado. Para alcançar estes objetivos, as empresas estão utilizando metodologias de desenvolvimento onde se torna necessário estimar o tempo das atividades do processo desde a fase de concepção até as fases de produção e testes. Isto tem exigido a adoção de métricas de estimativas de desenvolvimento de sistemas para medir o esforço e o tempo que um software produzirá.

O objetivo deste artigo é relatar um estudo de caso aplicando a métrica Pontos de Casos de Uso (PCU) numa empresa de software verificando o seu comportamento para, posteriormente, propor ajustes para sua melhor utilização. Esta empresa está iniciando a conversão dos seus sistemas para o ambiente Web utilizando uma metodologia de desenvolvimento para sistemas Web baseada no processo unificado de desenvolvimento e ainda não adotou nenhuma métrica para gerenciamento dos seus projetos. A métrica PCU foi escolhida pois permite fazer estimativas no início do projeto com base nos modelos de casos de uso construídos. Esta escolha deu-se devido as suas características intrínsecas serem as mais adequadas na estimativa dos tempos dos projetos orientados a objetos e com foco no ambiente WEB e por ser passível de medição na empresa de software estudada.

Nas seções seguintes é apresentada uma breve descrição sobre a gerência de projetos de software, o processo de desenvolvimento de software, métricas de estimativa de tamanho, a métrica PCU, a coleta de dados na empresa com a análise dos resultados, as conclusões e sugestões para trabalhos futuros.

2 O Processo de desenvolvimento de software

Existem tantas maneiras de desenvolver software quanto existem desenvolvedores. Entretanto, uma equipe de desenvolvimento de software precisa de uma estratégia unificada para desenvolver software. As metodologias de software definem uma maneira comum de encarar o desenvolvimento. Para atender a essa necessidade nasceu o Processo Unificado que é uma estrutura genérica de processo que pode ser customizada adicionando-se ou removendo-se atividades com base nas necessidades específicas e nos recursos disponíveis para um projeto. O Rational Unified Process (RUP) é um exemplo de versão customizada do Processo Unificado.

O Processo Unificado (SCOTT, 2003) faz uso extensivo da Linguagem de Modelagem Unificada - UML. Seus princípios fundamentais são:

- Dirigido por casos de uso: uma seqüência de ações executadas por um ou mais atores (pessoas ou entidades não-humanas fora do sistema) e pelo próprio sistema, que produz um ou mais resultados de valor para um ou mais atores. Os casos de uso não são selecionados isoladamente, os desenvolvedores devem começar pelos casos de uso chave e amadurecendo-os no decorrer do ciclo de vida do sistema.
- Centrado em Arquitetura: onde é descrita a visão do sistema como um todo. É o conjunto de tecnologias e ambientes sobre o qual o software será desenvolvido.
- Iterativo e Incremental: sendo que uma iteração é um mini-projeto que resulta em uma versão do sistema liberada interna ou externamente. Supõe-se que essa versão possua uma melhoria incremental sobre a anterior, motivo pelo qual é chamada de incremento.

O Processo Unificado divide-se em quatro fases ilustradas pela figura 1 e listadas a seguir:

- Concepção: o objetivo desta fase é estabelecer a viabilidade do sistema proposto.
- Elaboração: o objetivo desta fase é estabelecer a capacidade para a construção do novo sistema, dadas as restrições financeiras, de cronograma e de outros tipos de restrições com que o desenvolvimento se defronta.
- Construção: o objetivo da fase de construção é construir um sistema capaz de operar bem em ambientes beta de clientes.
- Transição: o objetivo desta fase é a de entregar o sistema completamente funcional aos clientes.

Segundo Scott (2003), cinco fluxos de trabalhos atravessam o conjunto das quatro fases do Processo Unificado. Cada fluxo de trabalho é um conjunto de atividades que vários membros executam. Eles estão descritos a seguir:

- Requisitos: visa construir um modelo de casos de uso que captura os requisitos funcionais do sistema que está sendo definido. Este modelo ajuda os interessados no projeto a chegar a um acordo sobre as capacidades do sistema e as condições que ele deve satisfazer.
- Análise: visa construir um modelo de análise, que ajuda os desenvolvedores a refinar e estruturar os requisitos funcionais capturados pelo modelo de casos de uso. Este modelo contém realizações de casos de uso mais apropriadas ao trabalho do projeto e de implementação do que casos de uso.
- Projeto: visa construir um modelo de projeto, o qual descreve as realizações físicas dos casos de uso a partir do modelo de projeto, o qual descreve as realizações físicas dos casos de usos a partir do modelo deste e do conteúdo do modelo de análise. O modelo serve como uma abstração do modelo de implementação.
- Implementação: visa construir um modelo que descreve como os elementos do modelo de projeto, como arquivos de código-fonte, bibliotecas de ligações dinâmicas e componentes executáveis (EJBs, por exemplo), são empacotados em componentes de software.

- Teste: visa construir um modelo de teste, que descreve como os teste de integração e de sistema exercitarão componentes executáveis a partir do modelo de implementação.

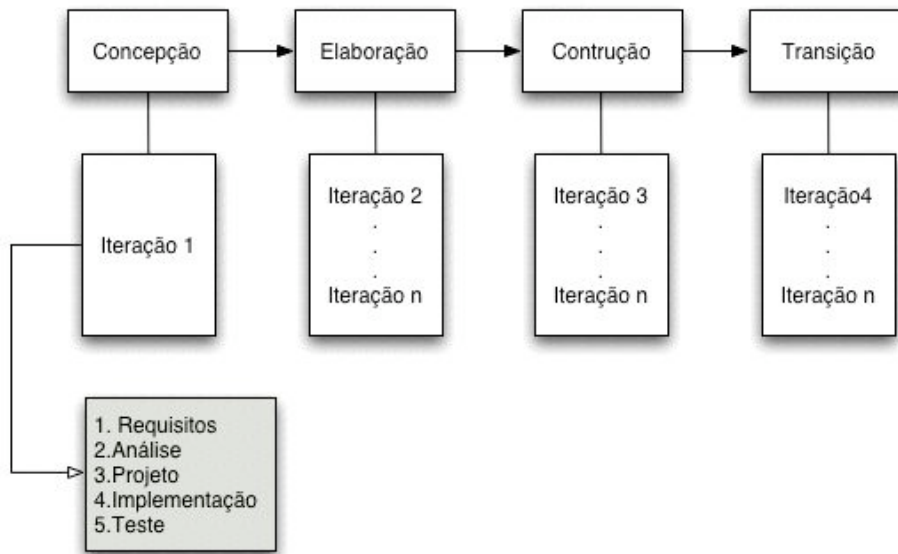


Figura 1: Mecanismo tradicional de iteração do processo unificado

Atualmente, o processo unificado vem sendo estudado, adaptado e adotado gradualmente por muitas empresas de software.

3 Métricas de estimativa de tamanho de software

Para Andrade e Oliveira (2004), métricas são metodologias de mensuração cujos principais objetivos, na área de análise de sistemas, são de estimar o tamanho de um software e auxiliar, como indicador, o gerenciamento dos projetos de desenvolvimento de sistemas. A estimativa de tamanho é uma das métricas mais utilizadas pois o tamanho do software tem impacto direto no esforço de desenvolvimento e na gestão do projeto. É um indicador da quantidade de trabalho a ser executado sendo possível definir esforço, prazo e custos necessários.

Segundo Pressmann (2002), a medição permite aos gerentes planejar, controlar, melhorar e aperfeiçoar o processo de desenvolvimento de software. Medição resulta em mudança cultural. Coletar dados, calcular e analisar métricas são três passos que devem ser implementados para iniciar um programa de métricas. O custo e o esforço necessários para construir software, mesmo em linhas de código produzidas são fáceis de coletar, desde que convenções específicas para a medição sejam estabelecidas antecipadamente. O software orientado a objetos é fundamentalmente diferente do software desenvolvido utilizando métodos convencionais. Por esse motivo as métricas para os sistemas orientados a objetos devem ser ajustadas considerando as características que a distinguem do software convencional. Existem várias métricas que podem ser utilizadas, dentre as quais encontram-se o número de linhas de código e a Análise de Pontos por Função (APF). Esta última muito utilizada nas organizações de software. Maiores informações sobre a Análise de

Pontos de Função podem ser obtidas em Vasquez, Simões e Albert (2003). A métrica Pontos de Casos de Uso foi escolhida por ser uma métrica baseada fortemente na UML e mais especificamente pelos casos de uso e ainda pelo fato da empresa estar iniciando o uso da UML.

3.1 A métrica de Pontos de Casos de Uso (PCU)

Os Pontos de Casos de Uso (PCU) foram criados em 1993 por Gustav Karner da empresa Objectory AB. Esta métrica permite fazer estimativas no início do projeto com base no modelo de casos de uso. A filosofia dos Pontos de Casos de Uso é baseada na definição da Análise de Pontos por Função (APF), na qual a funcionalidade vista pelo usuário é a base para a estimativa do tamanho do software (BELGAMO; FABBRI, 2004).

O processo de contagem dessa métrica consiste nos seguintes passos (MEDEIROS, 2004):

1. Relacionar os atores, classificá-los de acordo com seu nível de complexidade (simples, médio ou complexo) atribuindo respectivamente os pesos 1, 2 ou 3., conforme a tabela 1. Calcule o TPNAA (Total de Pesos não Ajustados dos Atores) somando os produtos da quantidade de atores pelo seu peso.

Complexidade do ator	Descrição	Peso
Simples	Muito poucas entidades de Banco de Dados envolvidas e sem regras de negócio complexas	1
Médio	Poucas entidades de Banco de Dados envolvidas e com algumas regras de negócio complexas	2
Complexo	Regras de negócios complexas e muitas entidades de Bancos de Dados presentes	3

Tabela 1: Classificação de atores (MEDEIROS, 2004)

2. Contar os casos de uso e atribuir o grau de complexidade sendo a complexidade baseada no número de classes e transações. Calcule o TPNAUC (Total de Pesos não ajustados dos casos de usos) somando os produtos da quantidade de casos de usos pelo respectivo peso conforme a tabela 2.

Tipo de Caso de Uso	Descrição	Peso
---------------------	-----------	------

Simple	Considerar até 3 transações com menos de 5 classes de análise	5
Médio	Considerar de 4 a 7 transações com 5 a 10 classes de análise	10
Complexo	Considerar de 7 transações com pelo menos de 10 classes de análise	15

Tabela 2: Classificação dos Casos de Uso (MEDEIROS, 2004)

3. Calcular PCU's não ajustados, também chamados de PCUNA, de acordo com a seguinte fórmula:

$$PCUNA = TPNAA + TPNAUC$$

4. Determinar o fator de complexidade técnica. Os fatores de complexidade técnica variam numa escala de 0 a 5, de acordo com o grau de dificuldade do sistema a ser construído. O valor 0 indica que a grau não está presente ou não é influente, 3 influência média e o valor 5 indica influência significativa através de todo o processo. Após determinar o valor dos fatores, multiplicar pelo respectivo peso ilustrado na tabela 3, somar o total e aplicar a seguinte fórmula:

$$\text{Fator de complexidade técnica (FCT)} = 0.6 + (0.01 * \text{Somatório do Fator técnico})$$

Descrição	Peso
Sistemas Distribuídos	2,0
Desempenho da aplicação	1,0
Eficiência do usuário final (on-line)	1,0
Processamento interno complexo	1,0
Reusabilidade do código em outras aplicações	1,0
Facilidade de instalação	0,5
Usabilidade (facilidade operacional)	0,5
Portabilidade	2,0
Facilidade de manutenção	1,0
Concorrência	1,0
Características especiais de segurança	1,0
Acesso direto para terceiros	1,0
Facilidades especiais de treinamento	1,0

Tabela 3: Fatores de complexidade técnica (MEDEIROS, 2004)

5. Determinar o fator de complexidade ambiental: os fatores de complexidade ambientais indicam a eficiência do projeto e estão relacionados ao nível de experiência dos profissionais. Esses fatores descritos na tabela 4 são determinados através da escala de 0 a 5, onde 0 indica baixa experiência, 3 indica média experiência e 5 indica alta experiência. Após determinar o valor de cada fator, multiplicar pelo peso e somar o total dos valores. Em seguida, aplicar a seguinte fórmula:
6. Fator de complexidade ambiental (FCA) = $1,4 + (-0,03 * \text{Somatório do Fator Ambiental})$

7. Calcular os PCU's ajustados: esse cálculo é realizado com base na multiplicação dos PCU não ajustados, na complexidade técnica e na complexidade ambiental através da seguinte fórmula:

$$PCUA = PCUNA * \text{Fator de complexidade técnica} * \text{Fator de complexidade ambiental}$$

Fator	Descrição	Peso
F1	Familiaridade com o processo de desenvolvimento de software	1,5
F2	Experiência na aplicação	0,5
F3	Experiência com OO, na linguagem e na técnica de desenvolvimento	1,0
F4	Capacidade do líder de análise	0,5
F5	Motivação	1,0
F6	Requisitos estáveis	2,0
F7	Trabalhadores com dedicação parcial	-1,0
F8	Dificuldade da linguagem de programação	-1,0

Tabela 4: Fatores de complexidade ambiental (MEDEIROS, 2004)

8. Calcular a estimativa de horas de programação. Karner, o criador da estimativa, sugere a utilização de 20 pessoas-hora por unidade de PCU. Schneider e Winters sugerem o seguinte refinamento :

X = total de itens de F1 a F6 com pontuação abaixo de 3

Y = total de itens de F7 a F8 com pontuação acima de 3

Se $X + Y \leq 2$, usar 20 como unidade de homens/hora

Se $X + Y = 3$ ou $X + Y = 4$, usar 28 como unidade de homens/hora

Se $X + Y \geq 5$, deve-se tentar modificar o projeto de forma a baixar o número, pois o risco de insucesso é relativamente alto.

Estimativa de horas = PCUA * pessoas hora por unidade de PCU

3.2 Estudo de caso

A empresa estudada é uma desenvolvedora de software corporativo, fundada há mais de quinze anos, de porte médio, com faturamento anual estimado em torno de R\$ 30 milhões e cerca de 300 funcionários. Possui uma área específica de pesquisa para o desenvolvimento de técnicas e de ferramentas de análise e programação, porém ainda não utiliza nenhuma métrica para as estimativas dos tempos de desenvolvimento.

Foram analisados os diagramas de casos de uso dos seguintes três projetos: Sistema de cálculo de Folha de Pagamento (Projeto 1), Sistema Contábil (Projeto 2) e o Sistema de Cartão-Ponto (Projeto 3). Em cada projeto foram realizadas as fases de concepção e a primeira iteração da fase de elaboração. Na fase de concepção somente foram elaborados os diagramas de nível 0 de apenas 1 módulo em cada projeto. Para a primeira iteração da fase de elaboração foram ampliados apenas alguns casos de uso considerados relevantes pelos analistas de cada projeto.

O índice do fator de complexidade ambiental obteve o mesmo valor para os 3 projetos pois os analistas possuíam o mesmo grau de experiência em UML, mesma familiaridade com processo unificado de desenvolvimento, mesma experiência em orientação a objetos, mesmo grau de motivação e mesmo conhecimento do ambiente de desenvolvimento, pois todos receberam um treinamento padronizado antes do início dos projetos. Os requisitos foram considerados estáveis pois trata-se da conversão de 3 sistemas já existentes em ambiente cliente/servidor para o ambiente Web.

O índice do fator técnico do projeto variou apenas nos itens referentes a complexidade de processamento, concorrência e acesso direto a terceiros. Todos os sistemas executam cálculos complexos e precisos, exigem máxima segurança e possuem mais de cem usuários acessando simultaneamente. Os pesos utilizados para os fatores técnicos e ambientais foram os sugeridos por Medeiros (2004).

As equipes dos respectivos projetos ainda não estão completas e a metodologia de desenvolvimento adaptada do processo unificado ainda está em teste, por isso o total de pessoas-horas por unidade de PCU considerado foi de 20 horas homem para uma primeira análise. As horas estimadas não foram divididas pelo total de membros da equipe. Os resultados desta pesquisa na fase de concepção foram ilustrados na tabela 5 e os da fase de elaboração na tabela 6.

Projetos	Atores	Casos de Uso	FCT	FCA	PCUNA	PCUA	Horas Estimadas
Projeto 1	4	1	1,00	0,81	22	17,93	358,60
Projeto 2	6	2	1,02	0,81	37	30,76	615,16
Projeto 3	7	2	1,03	0,81	39	32,74	654,77

Tabela 5: Estimativas da fase de concepção

Projetos	Atores	Casos de Uso	FCT	FCA	PCUNA	PCUA	Horas Estimadas
Projeto 1	4	5	1,00	0,81	72	58,68	1.173,60
Projeto 2	6	8	1,02	0,81	87	72,32	1,446,46
Projeto 3	7	4	1,03	0,81	64	53,72	1,074,50

Tabela 6: Estimativas da fase de elaboração

Verificou-se junto as equipes de desenvolvimento e aos coordenadores dos projetos que os tempos obtidos pela estimativa estavam muito acima dos obtidos em projetos semelhantes. Os coordenadores dos projetos identificaram que 20 horas/homem por total de tempo de unidade de PCU era um número muito alto e não representava corretamente uma boa média para todos os tipos de tempos por nível de complexidade de casos de uso. A metodologia de desenvolvimento da empresa utiliza uma camada de código que abstrai grande parte da geração de código básica dos seus sistemas, reduzindo a quantidade de horas/homem para realizar o desenvolvimento de casos de uso para aproximadamente 5 horas/homem em casos de uso simples, 9 horas/homem para casos de uso médios e 24 horas/homem para casos de uso complexos.

Foi decidido realizar uma nova estimativa nos três projetos utilizando os casos de uso da fase de elaboração por estar mais completa, desta vez ajustando-se os pesos dos casos de uso para simples=5, médio=10, complexo = 25 e modificar a quantidade de horas/homem para uma média de 10 horas/homem.

Os resultados obtidos foram os seguintes:

Projetos	Atores	Casos de Uso	FCT	FCA	PCUNA	PCUA	Horas Estimadas
Projeto 1	4	5	1,00	0,81	54	43,74	437,40
Projeto 2	6	8	1,02	0,81	76	61,50	615,60
Projeto 3	7	4	1,03	0,81	57	46,17	461,70

Tabela 7: Estimativas da fase de elaboração com os novos pesos ajustados

3.3 Análise dos resultados

O objetivo deste estudo foi verificar a viabilidade de aplicação da métrica PCU nos diagramas elaborados para uma primeira estimativa de horas necessárias para o desenvolvimento dos 3 projetos. O início da coleta destas estimativas se faz necessário para que a empresa comece a formar um histórico de dados através da comparação das horas estimadas com as horas efetivas ao final da fase de construção destes projetos.

Entre as duas fases houve um crescimento considerável de horas estimadas, pois na nova iteração foram melhor esboçados os requisitos nos diagramas de casos de uso. Diagramas de sequência e de classes também foram utilizados melhorando a identificação da complexidade dos casos de uso.

Segundo o coordenador geral das equipes de desenvolvimento, as quantidades de horas calculadas pela métrica PCU com o tempo padrão baseado em 20horas/homens não se aproximaram dos tempos normalmente auferidos pelas equipes de desenvolvimento através da experiência em programação em projetos anteriores. Nos projetos, os tempos calculados ficaram 70% abaixo dos tempos estimados pela métrica. Com o segundo cálculo, com os pesos dos casos de usos e horas/homem ajustados, as estimativas ficaram muito próximas da realidade atual das equipes de desenvolvimento.

Para uma estimativa mais precisa é necessário que todas as iterações da fase de elaboração estejam completas. Porém muitas empresas de software necessitam possuir o conhecimento do esforço ao final da fase de concepção para poderem alocar recursos, negociar prazos, fechar contratos com seus clientes, ou até mesmo para verificar se é viável ou não a continuidade do projeto.

Neste estudo houve dificuldade em identificar na coleta de dados os diagramas produzidos na fase de concepção. É necessário que empresa possua um registro dos diagramas produzidos ao final de cada fase ou mesmo a cada iteração dependendo dos objetivos que espera alcançar. Sugere-se que seja incluída uma atividade na metodologia de desenvolvimento da empresa de software para que o analista identifique e anote em cada diagrama de caso de uso a sua complexidade. O mesmo para os atores envolvidos no sistema.

Não foi possível a continuidade do levantamento de dados porque os projetos foram interrompidos, sendo sua continuidade prevista após esta pesquisa. A empresa estudada espera que após um ano de utilização da métrica PCU em seus projetos seja possível ao final da fase de elaboração um percentual de acerto de 90% em relação ao total de horas estimadas

4 Conclusões

A métrica PCU, assim como outros tipos de métricas, abordam as estimativas teóricas de tempo, e tem a finalidade de direcionar, viabilizar e ajudar controlar as fases do desenvolvimento de software.

O sucesso de um programa de métricas depende de uma série de fatores que variam desde a conscientização dos profissionais envolvidos até a existência de recursos técnicos e humanos necessários para a manutenção e monitoramento do programa. Um programa de medição traz benefícios para o programador a medida em que possibilita dimensionar melhor a carga de trabalho de forma a garantir a qualidade. Com esta informação o desenvolvedor poderá requerer mais recursos bem como dimensionar o valor pecuniário de seu trabalho além de estimar com mais precisão o tempo necessário para o desenvolvimento de suas tarefas. Para o empresário, este terá uma base de informações imprescindível para estimar seus custos e apreçar seus produtos, bem como estabelecer cronogramas com promessas de entrega de produtos em prazos possíveis de serem alcançados.

Em projetos orientados a objetos para que a estimativa de tamanho seja realizada com maior precisão desde o início do projeto pode-se utilizar as métricas Análises de Pontos de Função e Análise de Casos de Uso de forma combinada no momento em que elas são melhores aplicadas no processo de desenvolvimento. Conforme Andrade (2004) a confiança nas estimativas aumenta quando mais de uma forma de estimar é utilizada e, à medida que se obtém mais informações do domínio do software durante o processo de desenvolvimento do projeto, as estimativas serão melhores.

Este artigo foi elaborado para orientar os profissionais e iniciantes na atividade de desenvolvimento de sistemas sobre a importância da boa prática de controles durante a fase de desenvolvimento através da proposta de uma métrica já de domínio da área da ciência da computação.

Como sugestão para trabalhos futuros, a empresa poderá aplicar a métrica nos projetos pilotos nas próximas iterações até que a última iteração da fase de elaboração estiver concluída. Ao término da fase de construção será possível verificar os tempos de desenvolvimento reais em relação aos tempos estimados e propor novos ajustes na métrica PCU.

5 Referências

ANDRADE, Edméia Leonor Pereira; OLIVEIRA, Káthia Marçal. **Uso Combinado de Análise de Pontos de Função e Casos de Uso na Gestão de Estimativa de Tamanho de Projetos de Software Orientado a Objetos**. In III Simpósio Brasileiro de Qualidade de Software. Brasília: 2004

BELGAMO, Anderson; FABBRI, Sandra. **Um Estudo sobre a Influência da Sistematização da Construção de Modelos de Casos de Uso na Contagem dos Pontos de Casos de Uso**. In III Simpósio Brasileiro de Qualidade de Software. Brasília 2004

MEDEIROS, Ernani. **Desenvolvendo Software com UML 2.0**. São Paulo: ed. Makron Books, 2004.

PRESSMAN, Roger S. **Engenharia de software**. São Paulo : ed. McGraw-Hill, 2002.

SCOTT, Kendall. **O Processo Unificado Explicado**. Porto Alegre: ed. Bookman, 2003.

TAVARES, Helena Cristina A. B.; CARVALHO Ana Elizabete S.; CASTRO Jaelson F. B **Medição de Pontos por Função a Partir da Especificação de Requisitos**. Pernambuco : 2004

VAZQUEZ, Carlos Eduardo; SIMÕES, Guilherme Siqueira; ALBERT, Renato Machado. **Análise de pontos de função: medição, estimativas e gerência de projetos**. São Paulo : Érica, 2003.