

# Documentação Essencial para Manutenção de Software II

Sérgio Cozzetti Bertoldi de Souza<sup>1</sup>, Wesley Christian Gonçalves das Neves<sup>1</sup>,  
Nicolas Anquetil<sup>1</sup>, Káthia Marçal de Oliveira<sup>1</sup>

<sup>1</sup>Mestrado em Gestão do Conhecimento e Tecnologia da Informação  
Universidade Católica de Brasília (UCB) – Brasília – DF - Brazil

{sergiocozzetti, wchristiang}@yahoo.com.br,  
{anquetil, kathia}@ucb.br

**Resumo.** *A Engenharia de Software vem aprimorando métodos, técnicas e ferramentas em busca de qualidade e produtividade no desenvolvimento de software. Neste contexto, a documentação de software tem grande importância, pois pode influenciar tanto na qualidade quanto na produtividade. A falta ou o excesso de documentação pode colocar em risco o desenvolvimento e a manutenção do software. Encontrar um ponto de equilíbrio na quantidade de documentação gerada é de fundamental importância. Este artigo busca identificar na prática quais são os artefatos de software mais utilizados na atividade de obter o entendimento do sistema na fase de manutenção. O resultado desta pesquisa pode ser útil na definição de artefatos a serem gerados no processo de desenvolvimento de software cujo propósito da documentação seja o de dar suporte às atividades de manutenção.*

## 1. Introdução

Existe uma grande dificuldade das empresas que desenvolvem software cumprirem suas metas de prazo e orçamento. Segundo o Standish Group [15], somente 16,2% dos projetos de software terminaram dentro do prazo e orçamento estimados. Muitas vezes o prazo de entrega pode significar a sobrevivência ou não de uma organização. Como uma tentativa de minimizar este problema, estudos são realizados na área da Engenharia de Software. Neste contexto, a abordagem da documentação tem grande importância, no qual o prazo de entrega do projeto pode variar de acordo com a quantidade e qualidade dos artefatos de software a serem entregues.

Ambler [1] sugere duas razões básicas para documentar, uma serve para auxiliar a comunicação durante o projeto e outra para auxiliar o entendimento nas atividades de manutenção. A documentação é necessária quando é preciso estabelecer uma comunicação com uma equipe externa de trabalho. Ela serve como mecanismo de suporte à comunicação quando é combinada com reuniões, teleconferência, correio eletrônico e ferramentas colaborativas [1]. A documentação de software tem um efeito significativo no entendimento do programa [4, 17, 18]. Segundo Wong *et al* [19], mais de 50% do trabalho de evolução do software é dedicado ao entendimento do programa e a tarefa mais difícil da manutenção de software é o entendimento da estrutura do sistema [16].

Por outro lado, os métodos ágeis vêm ganhando popularidade por causa da rapidez de entrega. Um exemplo é o Extreme Programming (XP), que tem uma visão muito diferente e vem se tornando cada vez mais popular [3]. Ele confia exclusivamente na comunicação oral, testes e código fonte para comunicar intenção e estrutura [2]. Em outras palavras, se fielmente aplicado, não existe nenhum documento de análise e projeto [3].

Entretanto, a falta de documentação pode representar um risco para os projetos de software, principalmente na fase de manutenção. Ninguém pode garantir que o conhecimento, que está na cabeça dos programadores experientes, permanecerá vivo na equipe quando esta for dissolvida. Segundo Freeman e Munro [6], a falta de documentação ou a documentação incorreta afeta a compreensão do sistema, principalmente na atividade de análise de código.

Baseado nestas premissas, a questão é: qual é a documentação mínima (essencial) que sirva para o propósito de auxiliar o entendimento do sistema na fase de manutenção? Uma primeira pesquisa foi realizada como uma tentativa de identificar na prática quais são os artefatos de software essenciais com base na opinião de profissionais com experiência em manutenção de software [14]. Entretanto, este artigo visa validar as opiniões destes profissionais verificando a real utilização de cada artefato de software no dia a dia da manutenção. O resultado desta pesquisa será útil na definição de artefatos a serem gerados no processo de desenvolvimento de software cujo propósito da documentação seja o de dar suporte às atividades de manutenção.

Este artigo aborda alguns aspectos relacionados à documentação de software (seção 2). A seção 3 relata a estratégia utilizada para identificar na prática a real utilização dos artefatos de software na manutenção. Os dados da pesquisa são mostrados na seção 4. A discussão desses resultados, as conclusões e trabalhos futuros são abordados na seção 5.

## **2. Documentação de Software**

Documentação de software pode ser definida como um artefato cuja finalidade seja comunicar a informação sobre o sistema de software ao qual ele pertence [5]. Entretanto, é necessário distinguir entre modelos, documentos, código fonte e documentação. Segundo Ambler [1], do ponto de vista da modelagem ágil, um documento é qualquer artefato externo ao código fonte cujo propósito seja transmitir informação de uma maneira persistente. Modelo é uma abstração que descreve um ou mais aspectos de um problema ou de uma solução potencial para resolver um problema. Alguns modelos podem se tornar documentos, ou incluídos como parte deles, ou simplesmente serem descartados quando cumprirem seu papel. Código fonte é uma seqüência de instruções, incluindo os comentários que descrevem estas instruções, para um sistema de computador. O termo documentação inclui documentos e comentários de código fonte.

A documentação tem fundamental importância para a Engenharia de Software. Vários estudos tem sido realizados para minimizar os problemas em torno da documentação de software: documentação desatualizada e de baixa qualidade, processos de documentação dispendiosos e caros, documentação em abundância e sem propósito, dificuldade de acesso, entre outros. Alguns pesquisadores propuseram o uso de hipertexto para facilitar o acesso a

documentação. Tilley e Müller [17] propuseram combinar documentação e código fonte de uma maneira fácil e eficiente, utilizando ferramenta de hipertexto. Freeman e Munro [6] propuseram um sistema interativo em hipertexto com gráficos e texto. Rajlich [13] propôs a adoção de uma estratégia de redocumentação incremental e oportunística na qual a compreensão obtida na manutenção é armazenada, por meio de anotações, em um sistema de hipertexto baseado na Web. Tilley e Huang [18] definiram um modelo de maturidade para documentação de software com foco no produto (DMM – Documentation Maturity Model), baseado no CMM (Capability Maturity Model) do SEI (Software Engineering Institute), para estabelecer parâmetros de avaliação da qualidade da documentação. Ouchi [9] definiu uma estrutura de dados úteis para subsidiar um sistema de documentação voltado para manutenção de software e ressaltou a importância de métodos padronizados de documentação. Medina [10] descreveu os aspectos importantes para uma boa documentação e organização interna dos programas, como por exemplo, a utilização de comentários, identificação dos comandos, padronização de nomes de variáveis e espaçamento para enfatizar a estrutura lógica, no sentido de auxiliar o entendimento do programa. O HCi Journal [7] ressaltou que antes do início de cada projeto, se faz necessário um planejamento da documentação a ser utilizada, pois todo desenvolvimento tem características peculiares. Tilley [16] propôs um método flexível de identificação, documentação, representação e apresentação dos aspectos estruturais da arquitetura do software por meio da engenharia reversa chamada de “documenting-in-the-large”. A norma ANSI/ANS 10.3-1995 recomendou a divisão da documentação em quatro categorias: resumo, informação da aplicação, definição das funcionalidades e informação do programa [11]. Lethbridge *et al* [8] indicou a necessidade de empenhar na produção de um simples e poderoso formato e ferramenta de documentação, ao invés de forçar os engenheiros de software realizar um trabalho caro e ineficaz. Cioch et al [4] propuseram uma abordagem de documentação que considera o tipo de informação necessária para cada estágio de aprendizagem (recém-chegado, aprendiz, interno e experiente).

Pode-se notar que os estudos sobre documentação de software visam solucionar problemas de falta de atualização, dificuldade de acesso, falta de qualidade, desorganização, documentação desnecessária e indicam para soluções simples, com o apoio de ferramentas e que sirvam a um propósito.

### **3. Estratégia para Identificação da Documentação Essencial na Manutenção**

Numa primeira pesquisa [14] para identificar os artefatos de software que auxiliam no entendimento do sistema na fase de manutenção foi elaborado um questionário direcionado a profissionais com experiência em manutenção de software no qual cada entrevistado indicava o grau de importância de cada artefato para a manutenção. Esta primeira pesquisa foi respondida por 65 profissionais de manutenção.

O questionário utilizado nesta segunda pesquisa foi dividido em duas partes (Figura 1). A primeira com objetivo de caracterizar o entrevistado contendo nome, e-mail, tempo de experiência em manutenção, tempo de experiência no sistema em que está mantendo e tipo de manutenção realizada. A segunda parte foi reservada para o mantenedor indicar qual artefato ele utilizou para obter o entendimento do sistema no momento da manutenção.

Foram apresentados os artefatos mais comuns utilizados nas abordagens da análise estruturada e da orientação a objetos. Eles foram agrupados por fase do ciclo de vida tradicional [12], sendo que para cada abordagem existia um formulário específico. A lista de artefatos da análise estruturada teve como base métodos utilizados em algumas empresas nas quais os pesquisadores já trabalharam. Os artefatos da orientação a objeto tiveram como base a UML.

**Questionário: Artefatos Usados na Manutenção (ES)**

Este questionário tem o propósito de identificar quais são os artefatos de software (documentação) mais utilizados para obter o entendimento do sistema, na fase de manutenção, sob o ponto de vista dos mantenedores de software. Ele deve ser respondido por profissionais que atuam na atividade de manutenção de sistemas.

Em caso de dúvidas ou sugestões favor enviar email para: [sergioconnet@yahoo.com.br](mailto:sergioconnet@yahoo.com.br)

Caracterização do Mantenedor	
Nome (opcional):	
Email (opcional):	
Desp. recebido resultado da pesquisa?	( ) Sim ( ) Não
Qual a sua experiência em MANUTENÇÃO de sistemas (em anos)?	( ) 1 a 3 ( ) 3 a 5 ( ) 5 a 10 ( ) mais de 10 anos
Qual o tempo de experiência no sistema em que está dando manutenção (em meses)?	( ) 1 a 3 ( ) 3 a 6 ( ) 6 a 12 ( ) mais de 12 meses
Tipo de manutenção realizada no período da pesquisa:	( ) Corretiva - para corrigir erros não detectados na fase de teste
	( ) Adaptativa - para acompanhar a evolução inerente à computação
	( ) Perfectiva - para continuar satisfazendo as necessidades do usuário
	( ) Preventiva - para melhorar a confiabilidade ou manter a atualidade futura

**Questão**

Ao final de cada dia, marque com "X" quais artefatos de documentação você utilizou para obter o entendimento do sistema que você realizou manutenção.

**IMPORTANTE:** Marque somente os artefatos utilizados para obter o entendimento do sistema e NÃO os que você apenas teve que atualizar.

Observações:  
1 - Utilize o formulário apropriado para o tipo de abordagem no qual o sistema foi desenvolvido (Estruturada ou Orientação a Objetos);  
2 - Os artefatos estão agrupados por fase do processo de desenvolvimento para facilitar o preenchimento.

Escreva aqui a sua crítica, dúvida ou sugestão. Obrigada!!!

Formulário para sistemas desenvolvidos pela abordagem ESTRUTURADA									
Data de Início: ____/____/____					Data Fim: ____/____/____				
<b>Instruções para preenchimento:</b> 1) Marque com "X" quais artefatos de documentação você utilizou para obter o entendimento do sistema para cada manutenção realizada; 2) Caso o artefato não exista, marque somente a coluna "Não Existe". <b>IMPORTANTE:</b> Marque somente os artefatos utilizados para obter o entendimento do sistema e NÃO os que você apenas teve que atualizar.									
Relação de artefatos por fase	Não Existe	Manutenções							
		1	2	3	4	5	6	7	8
<b>Levantamento</b>									
Lista de Requisitos									
Diagrama de Contexto									
Descrição dos Requisitos									
Modelo Conceitual de Dados									
Uso de Sistema									
<b>Análise</b>									
Funções Derivadas dos Requisitos									
Diagrama Hierárquico de Funções									
Diagrama Fluxo de Dados									
Protótipo Não Funcional									
Modelo Lógico de Dados (MLD)									
Diagrama de Dados									
<b>Projeto</b>									
Modelo de Arquitetura									
Diagrama Geral de Transações									
Especificação de Componentes									
Modelo Físico de Dados									
Protótipo Funcional									
<b>Implementação</b>									
Comentários do Código Fonte									
Código Fonte									
<b>Teste</b>									
Plano de Teste Unitário									
Plano de Teste Sistema									
Plano de Teste Homologação									
<b>Implantação</b>									
Plano de Migração de Dados									
Plano de Implantação									
Manual do Usuário									
<b>Outros não identificados acima</b>									

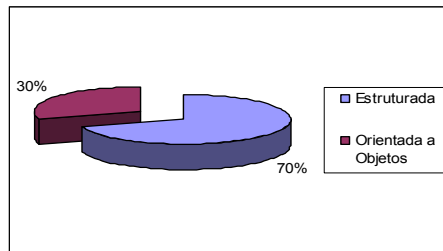
Figura 1. Questionário para identificar os artefatos usados na manutenção.

#### 4. Análise e Resultados da Pesquisa

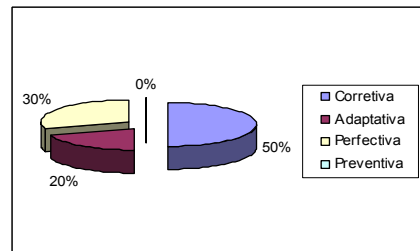
O questionário foi distribuído para profissionais mantenedores de software em várias empresas que mantêm software. Até o momento, sete profissionais de três empresas responderam o questionário, totalizando dez manutenções realizadas nos meses de julho e agosto de 2004. Das dez manutenções observadas, somente três foram relativas à abordagem orientada a objetos (Figura 2). Este número foi considerado insuficiente para se chegar a alguma conclusão sobre esta abordagem e não iremos considerá-la neste artigo.

Grande parte das manutenções realizadas foi corretiva (50%), seguida da perfectiva (30%) e adaptativa (20%). Não foi constatada nenhuma manutenção preventiva, conforme é mostrado na Figura 3.

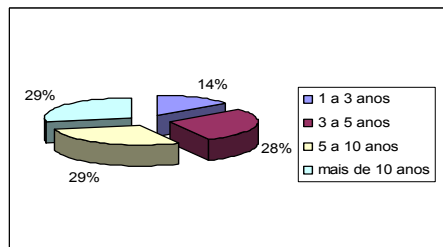
A Figura 4 mostra um grande nível de experiência dos entrevistados, sendo que a maioria deles (58%) possui mais de cinco anos de vivência na manutenção de software. Entretanto, nem todos os participantes têm muita experiência com o sistema que estavam mantendo. Trinta por cento dos mantenedores tem menos de três meses de experiência (Figura 5), sendo que nestes casos a necessidade de entendimento era maior.



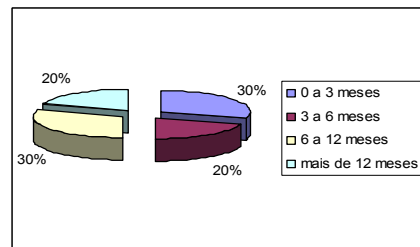
**Figura 2.** Abordagem dos sistemas mantidos



**Figura 3.** Tipo de manutenção



**Figura 4.** Tempo de experiência em manutenção de software



**Figura 5.** Tempo de experiência nos sistemas mantidos

## Uso dos artefatos de documentação na manutenção de software

A Tabela 1 mostra os resultados resumidos das duas pesquisas sobre a documentação essencial para manutenção. Os dados sobre a importância dos artefatos da análise estruturada foram obtidos na pesquisa realizada por SOUZA et al [14]. Os dados relativos à utilização efetiva dos artefatos tiveram como base os dados coletados na segunda parte deste questionário. É importante ressaltar que foi considerado o uso do artefato no momento de obter o conhecimento do sistema para realizar a manutenção, bem como se o artefato existe na documentação disponível de cada sistema mantido.

**Tabela 1. Resultados das duas pesquisas sobre documentação essencial para manutenção na abordagem estruturada**

Abordagem Estruturada		Importância			Uso	
Artefatos	Sem Importância	Pouco Importante	Importante	Muito Importante	Uso	Existência

Código Fonte	0,0%	0,0%	6,2%	93,8%	100,0%	100,0%
Comentários do Código Fonte	0,0%	7,6%	15,2%	77,3%	100,0%	100,0%
Modelo Lógico de Dados	0,0%	4,7%	23,4%	71,9%	100,0%	85,7%
Descrição dos Requisitos	4,8%	11,1%	23,8%	60,3%	40,0%	71,4%
Modelo Físico de Dados	0,0%	3,1%	37,5%	59,4%	85,7%	100,0%
Lista de Requisitos	9,7%	12,9%	25,8%	51,6%	40,0%	71,4%
Dicionário de Dados	1,6%	17,5%	33,3%	47,6%	85,7%	100,0%
Modelo Conceitual de Dados	6,5%	12,9%	35,5%	45,2%	40,0%	71,4%
Plano de Testes de Homologação	9,8%	14,8%	31,1%	44,3%	0,0%	42,9%
Manual do Usuário	9,5%	17,5%	34,9%	38,1%	16,7%	85,7%
Plano de Testes de Sistema	6,3%	17,5%	39,7%	36,5%	42,9%	100,0%
Plano de Implantação	7,8%	15,6%	42,2%	34,4%	33,3%	85,7%
Plano de Testes Unitário	8,1%	21,0%	37,1%	33,9%	50,0%	85,7%
Protótipo Funcional	11,5%	21,3%	36,1%	31,1%	20,0%	71,4%
Diagrama de Fluxo de Dados	8,1%	16,1%	45,2%	30,6%	0,0%	42,9%
Modelo de Arquitetura	8,8%	26,3%	35,1%	29,8%	0,0%	71,4%
Especificação de Componentes	6,8%	16,9%	47,5%	28,8%	0,0%	57,1%
Plano de Migração de Dados	11,7%	18,3%	41,7%	28,3%	0,0%	57,1%
Diagrama Hierárquico de Funções	8,6%	22,4%	46,6%	22,4%	0,0%	42,9%
Diagrama de Contexto	6,7%	35,0%	36,7%	21,7%	0,0%	57,1%
Funções Derivadas dos Requisitos	8,5%	31,9%	38,3%	21,3%	0,0%	57,1%
Protótipo não Funcional	13,8%	22,4%	43,1%	20,7%	0,0%	57,1%
Glossário	7,9%	33,3%	38,1%	20,6%	20,0%	71,4%
Diagrama Geral de Transações	10,2%	26,5%	42,9%	20,4%	0,0%	57,1%

Várias análises puderam ser feitas a partir dos resultados obtidos em cada pesquisa bem como na comparação entre ambas. Na pesquisa sobre a importância dos artefatos [14] pôde-se observar o quanto a documentação é considerada importante para a manutenção na opinião dos profissionais da área. Na pesquisa sobre o uso efetivo dos artefatos pôde-se notar o quanto cada artefato é utilizado pelo mantenedor no momento da obtenção do entendimento do sistema e novamente constatar a importância da documentação.

Os artefatos mais utilizados foram “Código Fonte”, “Comentários do Código Fonte” e Modelo Lógico de Dados, todos com 100% de utilização. Logo em seguida vem “Dicionário de Dados” e “Modelo Físico de Dados”, ambos com 85,7%. Com um pouco menos de indicação seguem-se os planos de teste, “Plano de Teste Unitário” (50%) e “Plano de Teste de Sistema” (42,9%).

Os artefatos considerados muito importantes na primeira pesquisa [14] foram “Código Fonte” com 93,8% de indicação, “Comentários do Código Fonte” (77,3%), “Modelo Lógico de Dados” (71,9%), “Descrição dos Requisitos” (60,3%), “Modelo Físico de Dados” (59,4%), “Lista de Requisitos” (51,6%) e “Dicionário de Dados” (47,6%).

## 5. Discussão e Conclusão

Os resultados da pesquisa constatarem a importância da documentação na manutenção de software, pois os artefatos com menor índice de importância tiveram mais de 50% de

indicação dos mantenedores. Quanto ao grau de utilização, dos vinte e quatro artefatos indicados, quatorze foram utilizados pelos mantenedores. Este fato comprova o risco que representa a falta de documentação para a manutenção de software. Por outro lado, nem todos os documentos foram utilizados pelos mantenedores para buscar o entendimento do sistema. Isto reforça também a necessidade de se ter um propósito para o qual a documentação é criada e mantida.

No entanto, o propósito deste artigo foi o de identificar na prática quais são os artefatos de software essenciais na atividade de obter o entendimento do sistema na fase de manutenção, confrontando a opinião dos profissionais da área com o grau de utilização dos artefatos de documentação. Em geral, pôde-se constatar que os artefatos indicados como mais importantes pelos profissionais são também os mais utilizados na prática pelos mantenedores. O artefato mais importante sem nenhuma dúvida é o código fonte, que juntamente com os comentários nele existentes reforçam a importância da adoção de padrões de codificação e de organização interna do código conforme relatado por Medina [10]. Ambos foram usados por todos os entrevistados. Outros artefatos que foram bastante utilizados foram o modelo de dados (físico e lógico) e o dicionário de dados, o que comprova a importância atribuída a eles e ressalta a necessidade desses modelos estarem atualizados e disponíveis no momento da manutenção. Foi dada uma importância muito grande a descrição dos requisitos na primeira pesquisa, no entanto sua utilização prática ficou um pouco abaixo média (40%). Por outro lado, os planos de testes unitário e de sistema tiveram uma utilização prática razoável (50% e 42,9% respectivamente) enquanto não eram considerados muito importantes pela maioria dos profissionais na primeira pesquisa. Pôde-se notar também que a maioria dos artefatos considerados menos importantes não foi utilizada pelos mantenedores.

O próximo passo da pesquisa será o de obter mais dados para validar os resultados obtidos na abordagem estruturada e que seja possível aplicar na abordagem orientada a objetos também. Outro passo será o de verificar na literatura a existência de algum artefato que seja essencial para a manutenção de software.

O questionário utilizado nesta pesquisa levou em consideração as abordagens estruturada e orientada a objetos, no entanto, o questionário pode ser adaptado e aplicado para outras abordagens ou métodos específicos alterando-se apenas a listas de artefatos.

Os dados desta pesquisa podem ser úteis para várias análises em torno dos artefatos de documentação de software no sentido de buscar o equilíbrio na produção de artefatos. A partir destes resultados pode ser possível a identificação de quais artefatos podem ser considerados em um método cujo propósito da documentação seja o de gerar artefatos úteis para dar suporte a manutenção de software.

## **Referências Bibliográficas**

1. AMBLER, Scott W. Agile Documentation. 2001-2004, The Official Agile Modeling (AM) Site, 2001, Disponível em : <<http://www.agilemodeling.com/essays/agileDocumentation.htm>>, Acesso em: 02 abr. 2001.
2. BECK, Kent. Extreme Programming Explained: Embrace Change. Boston: Addison-Wesley, 2000.

3. BRIAND, Lionel C.. Software Documentation: How Much is Enough?, Proceedings of the 7th European Conference on Software Maintenance and Reengineering (CSMR'03), IEEE, 2003, p.13.
4. CIOCH, Frank A.; PALAZZOLO, Michael; LOHRER Scott. A Documentation Suite for Maintenance Programmers, Proceedings of the International Conference on Software Maintenance (ICSM '96), IEEE, 1996, p.286-295.
5. FORWARD, Andrew. Software Documentation – Building and Maintaining Artefacts of Communication, 2002. Teste de Mestrado, Universidade de Ottawa, Ottawa, Toronto, Canadá.
6. FREEMAN, Robert M.; MUNRO, Malcolm. Redocumentation for the Maintenance of Software, Proceedings of the 30th annual Southeast regional conference, ACM, 1992, p.413-416.
7. HCI JOURNAL. What to put in software maintenance documentation. HCl Consulting, 2001, Disponível em : <<http://www.hci.com.au>>, Acesso em: 06 nov. 2003.
8. LETHBRIDGE, Timothy C.; SINGER Janice; FORWARD, Andrew. How Software Engineers Use Documentation: The State of the Practice, IEEE Software, Novembro, 2003, IEEE, p.35-39.
9. OUCHI, Miheko L.. Software Maintenance Documentation, Proceedings of the 4th annual international conference on Systems documentation (SIGDOC'85), Nova Iorque, EUA, ACM Press, 1985, p.18-23.
10. MEDINA, Enrique A.. Some Aspects of Software Documentation, Proceedings of the 3rd annual international conference on Systems documentation (SIGDOC'84), Cidade do México, 1984, p.57-59.
11. PHONA, Vir. A Standard for Software Documentation, IEEE Computer, Outubro, 1997, p.97-98.
12. PRESSMAN, Roger S. Software Engineering: a practitioner's approach. Nova Iorque: McGraw-Hill, 5ª Edição, 2001.
13. RAJLICH, Václav. Incremental Redocumentation Using the Web, IEEE Software, Setembro/Outubro 2000, p.102-106.
14. SOUZA, Sérgio C. B. de; NEVES, Wesley C. G. das; ANQUETIL, Nicolas; OLIVEIRA, Káthia M. de. Investigação da Documentação de Maior Importância para Manutenção de Software, artigo aceito na 4ª Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC'04), Outubro, 2004.
15. THE STANDISH GROUP. The Standish Group Report - CHAOS. The Standish Group, 1995, Disponível em: <<http://www.scs.carleton.ca/~beau/PM/Standish-Report.html>>, Acesso em: 07 abr. 2004.
16. TILLEY, Scott R.. Documenting-in-the-large vs. Documenting-in-the-small, Proceedings of the 1993 conference of the Centre for Advanced Studies on Collaborative research: distributed computing - Volume 2 (CASCON'93), Toronto, Ontário, Canadá, 1993, p.1083-1090.
17. TILLEY, Scott R.; MÜLLER, Hausi A.. INFO: A Simple Document Annotation Facility, Proceedings of the 9th annual international conference on Systems documentation (SIGDOC'91), ACM Press, 1991, p.30-36.
18. TILLEY, Scott R.; HUANG, Shihong. Towards a Documentation Maturity Model Proceedings of the 21st annual international conference on Documentation (SIGDOC'03), San Francisco, Califórnia, EUA, Outubro, 2003, p.93-99.
19. WONG Kenny, TILLEY, Scott R.; MÜLLER, Hausi A.; STOREY Margaret-Anne D.. Structural Redocumentation: A Case Study, IEEE Software, Janeiro, 1995, p.46-54.