IBM DEVELOPMENT CONFERENCE 2008

ONDE PROFISSIONAIS DE TI SÃO HERÓIS

DOCUMENTAÇÃO DE SISTEMAS - IMPORTÂNCIA, APLICAÇÃO E USO PARA O CICLO DE VIDA DE SOFTWARES

Denis Mello
Consultor Sênior - BearingPoint

O que é Documentação de Sistemas?

- Objetivos da Apresentação
  - Introdução
  - Motivação
- O que é a Documentação de Sistemas?
  - Tipos de Documentação;
  - Pragmatismo;
  - Importância;
  - RUP e sua contribuição
  - Estudo de Caso;
  - Considerações Finais;
  - Fundamentação Teórica.

Introdução

- “Documentação de usuários, sistemas e processos tornaram-se mais necessários uma vez que companhias terceirizam o desenvolvimento de *softwares*. Os padrões de documentação atuais são insuficientes para necessidades emergentes. (Gartner Group, 2004).

- “*It's harder to know where you need to go, if you have no record of where you have been.*” *(IT* Project Management Tools Templates Workbooks Guides and Methodologies*).*

Motivação

- Os fatores motivadores para apresentar este tema são:
  - A falta de padrões internacionais estabelecidos para documentação;
  - A lacuna no entendimento e aplicação de conceitos importantes;
  - Mistificação de seu propósito e aplicabilidade;
  - A grande heterogeneidade de formatos e usos nas empresas;
  - A forte pressão que Gestores de TI sofrem em adeqüar suas Equipes a Padrões de Mercado, Metodologias e incorporar um Processo que suporte o Ciclo de Vida de Sistemas

O que é Documentação de Sistemas?

- **Necessidade ou Preciosismo?**
  - A documentação é parte, ou deveria ser, integrante de qualquer sistema ou programa criado;
  - É um artefato cuja finalidade é comunicar a informação sobre o sistema/software ao qual ele pertence
  - Artefatos que mostram o sistema sob diversas perspectivas, contendo informações suficiente para a compreensão de sua abrangência, bem como permitir a continuidade no suporte e operação, independente da equipe que esteja à frente
  - Sua importância é cada vez maior em função do amadurecimento de **Metodologias e Melhores Práticas** (RUP, XP, ITIL, COBIT) e mesmo regulamentações como a Sarbanes-Oxley. O que todas têm em comum? A necessidade de se documentar cenários, processos e sistemas

Tipos de Documentação

Functional Specification	This document describes the business requirements. It describes the possible effects of a software system, in other words, *what* the system must accomplish
Technical Specification	This document describes the approach for addressing both functional and non-functional requirements. It describes "*how to do it*"
Installation Instruction	This document states the instructions for installing and/or deploying the application.
Operations Runbook	This document describes the particular instructions for operating and keeping the applications up and running. It also states the actions to be taken in case exceptions occur.
User Training Material	The description of the most relevant (usually business process-related) information is the scope of this document.
Test Plan	A test plan is a systematic approach to testing a system. The plan typically contains a detailed understanding of what the eventual workflow will be. It includes Test Cases, Criteria to Pass/Fail and others

Pragmatismo

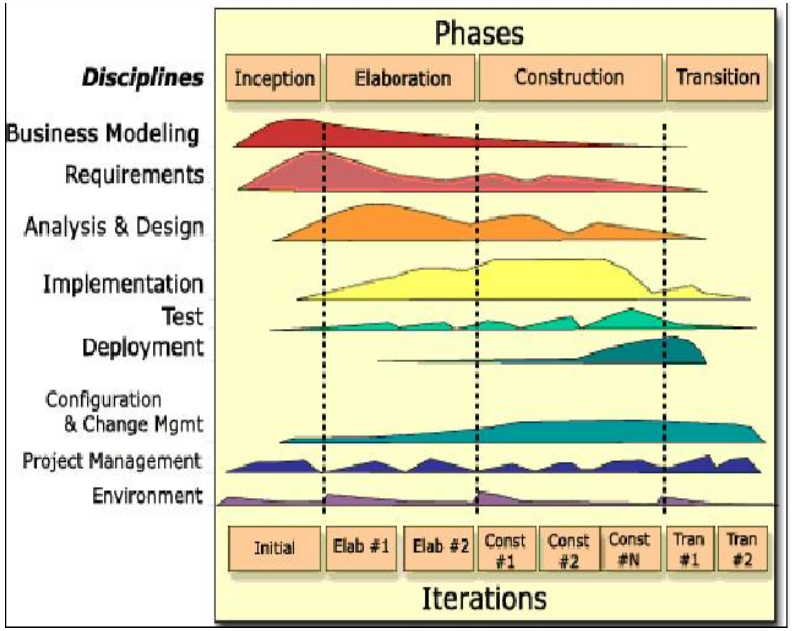
- Para muitos, a criação de documentação é a parte mais díficil e complicada para se enfrentar em todo o processo de criação de um *software*, seja pela necessidade de escrever várias e várias páginas de texto, gráficos e desenhos ou ainda pela necessidade de largar aquilo que se aprendeu (programar) para fazer aquilo que não sabe bem (redigir).
  - **Desenvolver documentação é um processo que demanda tempo!**
  - **Desenvolvê-la é algo muito oneroso.**
  - **Por que desenvolver se tenho profissionais na equipe que sabem como desenvolver?**
  - **O que uma DS deve conter?**
  - **Como desenvolver uma? Por onde começo?**

Importância

- Há algumas razões válidas para manter sistemas e respectiva documentação:
  - Passar em uma auditoria de TI (operações, políticas e procedimentos);
  - Criar uma gravação com acuracidade do design, manutenção e atualização dos sistemas;
  - Fornecer ao time de TI uma **base de conhecimento** sobre seus sistemas, procedimentos operacionais, administração, suporte e manutenção.
  - Assegurar consistência e segurança frente à mudanças de equipe, transições organizacionais, processos de terceirização ou fusão
  - Minimizar ou eliminar o risco de esforços com mecanismos de **engenharia reversa** durante a manutenção dos sistemas;
  - Reduzir a ocorrência de "*job security*" nas empresas.

RUP e sua contribuição

- O Ciclo de Vida do RUP engloba artefatos essenciais que descrevem o sistema sob várias perpectivas. O RUP basicamente enfoca:
  - Requerimentos
  - Arquitetura e Design;
  - Testes
  - Documentação de Usuário
- Alguns dos princípios fundamentais do RUP (Desenvolvimento Iterativo, Gerenciamento de Requerimentos e verificação da qualidade), asseguram o ciclo de vida de sistemas através de artefatos concretos, sobretudo documentos;
- O RUP utiliza a Linguagem Unificada de Modelagem **(UML)** para especificar, modelar e documentar artefatos. A **UML** é um padrão definido pelo OMG e tem se tornado o padrão empresarial para a modelagem orientada a objetos;



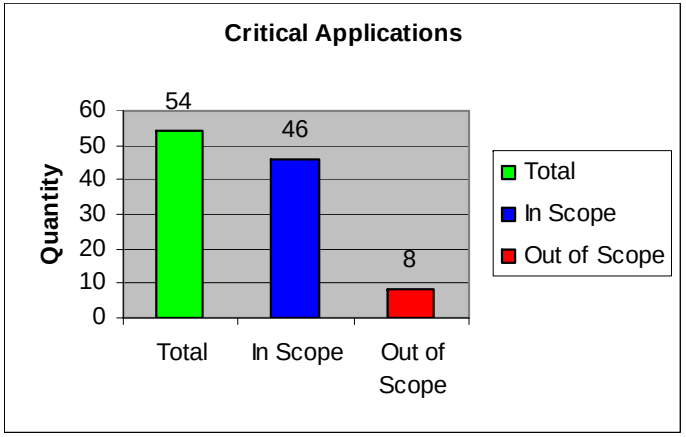
- O RUP, portanto, suporta fortemente a documentação, garantido em cada fase do ciclo de vida do desenvolvimento do sistema e análise contínua, a aplicação de artefatos que capturam as informações do sistema sob dada perspectiva.

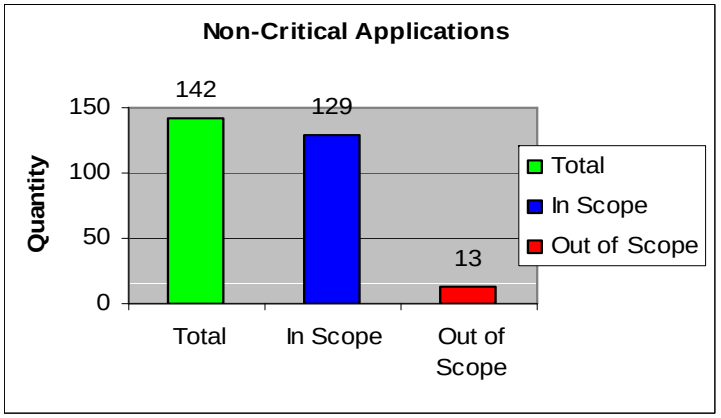
Case Presentation

- **Executive Summary:** The company (a global manufacturer) was working on a Global initiative to improve the Selection, Contract and Management of outsourcing for Systems and Technology Services.
- The company did not want new vendors to determine the level of risk they can afford when looking for new contracts. The organization was completely aware that having a considerably high amount of transitions at the same time involves by itself an elevated risk.
- The company knew that a **qualified and complete documentation to operate a systems reduces significantly the level of risk in the event of a service supplier transitioning**.

Scope

- The engagement was in charge of assessing the existing technical documentation and proposing a common standard of documents.
- For all systems, the BearingPoint team made use of six types of documents to assess the entire amount of applications and systems over LA, Middle Africa and Brazil (total of 10 countries);
  - More than 15 Service Supliers involved:
  - More than 2000 documents assessed during the project





Findings and Main Considerations

- The six types of documents were far enough to support the whole process and also to cover and analyze any type of document provided, don't matter the type of application or infrastructure it supported;
- Lack of documentation standard.
- Outdated documentation;
- Misconception of documentation requirements;
- Different file formats and layouts not in accordance with the SOW and SLA not followed by Service Suppliers in terms of Documentation Development;
- Poor quality of documentation content in many sittuations;
- Risks:
- Steep learning curve for mastering the applications.
- Time-consuming knowledge transfer process;
- Inability to widespread no sensitive information and leverage the existing documentation;
- Requiring costly reverse engineering activities for leveraging or even maintaining the life cycle of the applications.

Considerações Finais

- É comum uma organização gastar de 20 a 30% de todo o esforço de desenvolvimento na elaboração de documentos
- De fato, deve-se considerar que a elaboração de uma documentação de qualidade é tão importante quanto a qualidade do *software* em si
- A documentação deve ser usada como uma ferramenta de *marketing* para compartilhar informações de valor e fortalecer as linhas de comunicação entre TI e usuários finais.
- **Propriedades Importantes:**
  - **Rastreabilidade –** Onde posso encontrar o código, cenário ou diagrama que implementa um elemento particular ou onde nos requerimentos é necessário estar um código particular ou propriadade necessária de implantação?
  - **Coerência** – Posso navegar de seção a seção e de um tipo de representação a outra e manter a rastreabilidade?
  - **Consistência** — As representações usadas estão em concordância?
  - **Completa** — Todo o código ou representação dos principais cenários descritos? Existem os elementos da documentação que são necessários para o código?
  - **Precisão** — Como os requerimentos e código evoluem através de versões sucessivas, estão o documento ou elementos sendo atualizados?

Fundamentação Teórica

- **Termos e Conceituação Técnica***:* W3C, IEEE 100, Olsson e Runeson (2002), Rational Unified Process (versão 2003.06.15) e UML

- **Engenharia de Software e Orientação a Objetos:** Sommerville (2001), Booch (1994 e 1995) e Pressman (2005).

- Gartner Group e R. S. Pressman