

Influência dos Requisitos na Qualidade

Qualidade de Software (2011.0)

Prof. Me. José Ricardo Mello Viana

Conteúdo

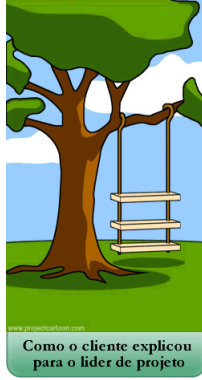
1. Introdução
2. Requisitos de software
3. Documento de requisitos
4. Tipos de requisitos
5. Técnicas de levantamento de requisitos
6. Qualidade de Requisitos
7. Perspective-Based Reading
8. Diálogo entre cliente e desenvolvedor
9. Ferramentas

Introdução

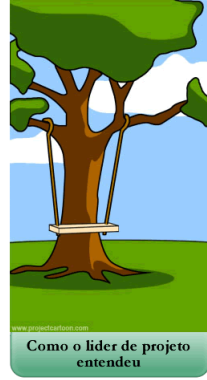
- Definir com precisão os requisitos permite direcionamento claro
 - Dos recursos da empresa
 - Da energia da equipe de desenvolvimento
- Sem essa definição
 - Perde-se tempo
 - Mais erros são cometidos
 - Qualidade do produto é incerta

Introdução

- Comparação com a instalação de um balanço em uma árvore



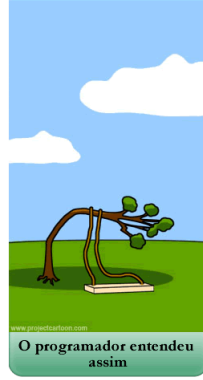
Como o cliente explicou para o líder de projeto



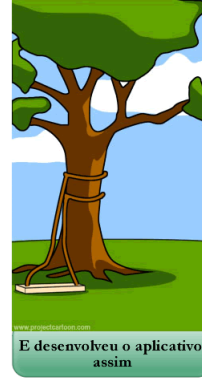
Como o líder de projeto entendeu



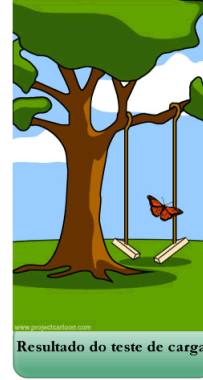
O analista especificou assim



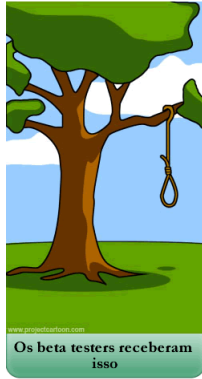
O programador entendeu assim



E desenvolveu o aplicativo assim



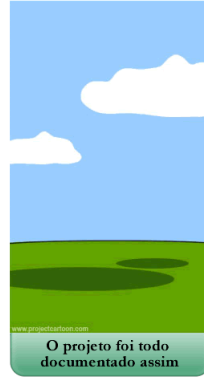
Resultado do teste de carga



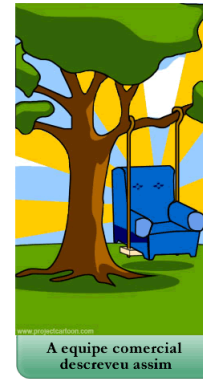
Os beta testers receberam isso



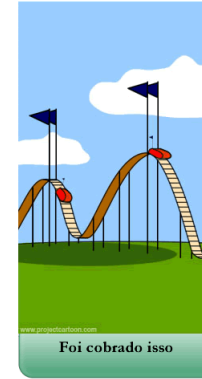
Como os patches foram aplicados



O projeto foi todo documentado assim



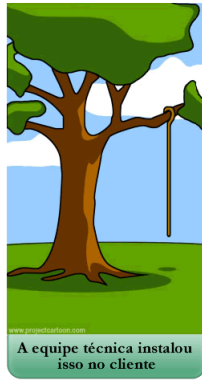
A equipe comercial descreveu assim



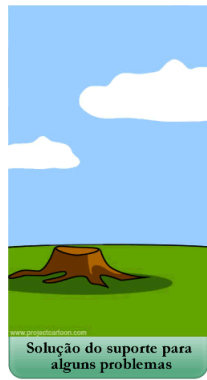
Foi cobrado isso



iSwing
E o software foi anunciado assim



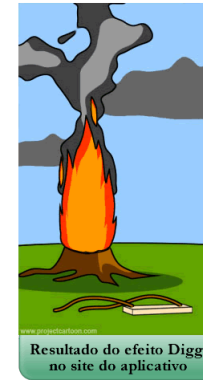
A equipe técnica instalou isso no cliente



Solução do suporte para alguns problemas



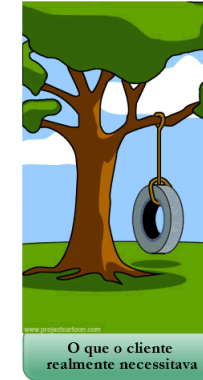
Quando ele foi entregue



Resultado do efeito Digg no site do aplicativo



A versão Open Source



O que o cliente realmente necessitava

Introdução

- Cliente procura empresa precisando de um software
- Analista de sistemas o entrevista e colhe dados
 - Acrescenta um toque pessoal, para cobrir lacunas
- Analista expõe suas anotações à equipe
 - Gerente decide fazer algumas modificações
- Analista percebe que algumas mudanças foram prejudiciais
 - Adapta com base em suas próprias experiências
- Projeto é entregue ao programador
 - Efetua ajustes de modo a reutilizar código
- Considerando que isso seja passado de forma oral
 - Pode ser esquecido ou entendido de diversas maneiras

Requisitos de software

- Descrições sobre comportamento, funções e operações que deve realizar
 - Levantamento, análise, especificação, rastreamento e validação
- Principais causas de falhas em projetos
 - Dificuldade em entender o que o usuário quer
 - Descrições incompletas
 - Mudanças não controladas
- Forma que as operações são realizadas
 - Visão do cliente e visão do programador

Requisitos de software

- Num software operações e dados estão relacionados
- Durante o ciclo de vida do software os requisitos sofrem influência de diferentes pessoas
 - *Stakeholders*: pessoas que tem controle sobre a especificação
 - *Operadores*
 - *Gerentes*
 - *Compradores*
 - *Leis e regulamentos*
 - *Analistas*
 - *Programadores*
- Além de pessoas, podem influir
 - Natureza da aplicação, tecnologia disponível

Requisitos de software

- Lidando com o desconhecido
 - Deve haver atenção com requisitos não especificados, mas que fazem parte do produto
 - Automóveis devem ter janelas, softwares interfaces inteligíveis
 - Diagrama de visibilidade dos requisitos
 - Requisitos óbvios
 - Ausência de bugs, facilidade de operar
 - Cliente é a fonte de informações sobre seu negócio
 - Podem ser extremamente práticos ao descrever
 - Ou não conseguem (não querem)
 - Requisitos documentados: explícitos
 - Requisitos não especificados, mas necessários: implícitos

Documento de requisitos

- Escrever de forma correta e concisa é fundamental para sua compreensão
 - Não basta apenas saber programar para ser bom profissional
- Basta uma pequena inconsistência para levar a problemas
- Boas práticas
 - Alocar tempo: não ter pressa ao escrever o documento
 - Consistência: requisitos devem ser entendidos de uma só forma por todos que o leiam (evitar sinônimos)
 - Concisão: frases e parágrafos curtos, evitando adjetivos

Documento de requisitos

- Deve conter
 - Serviços e funcionalidades que o software deve ter
 - Restrições de operação
 - Propriedades gerais do software
 - Requisitos de hardware
 - Definição de quais outros softwares devem ser integrados
- Padrão IEEE-Std-830-1998
 - Introdução, propósito do documento, escopo do produto, definições e abreviações, referências, visão geral do restante do documento, descrição geral, perspectiva do produto, funções do produto, características do usuário, restrições gerais, suposições e tendências, requisitos adiáveis, requisitos específicos, apêndices, índice

Tipos de requisitos

- Requisitos funcionais
 - Funcionalidades que se espera que o software tenha
 - Deve ser completa e consistente
 - Todas as funções requeridas pelo usuário devem estar definidas
 - Não devem ser contraditórias
 - Para softwares grandes é quase impossível
 - Diferentes pontos de vista
 - Devem ser desvinculados o máximo possível da tecnologia

Tipos de requisitos

- Requisitos não funcionais
 - Restrições ao software de forma geral
 - Ex: confiabilidade de um sistema de controle de voo
 - É difícil verificá-los
 - Ex: capacidade de detecção, facilidade de uso, rapidez de resposta
 - Estão abertos a interpretação
 - Idealmente devem ser expressos quantitativamente
 - Usar métricas
 - Transações processadas por segundo, tamanho em bytes do sistema, memória necessária
 - Podem entrar em conflito e interagem com requisitos funcionais
 - Ex: desempenho, verificação, portabilidade, recursos, interface, eficiência, qualidade, segurança, interoperabilidade, robustez, confiabilidade, manutenibilidade

Técnicas de levantamento de requisitos

- Não existe um processo ideal
- Muitos acham que é necessário começar a programação o mais rápido possível
 - Boa especificação de requisitos custa tempo e dinheiro
 - No entanto, sua ausência pode custar ainda mais dinheiro
- Possíveis problemas
 - *Stakeholders* podem não estar de total acordo
 - Ocorrem decisões unilaterais
 - Gerentes podem impor requisitos que não sejam ideais
- Processo cooperativo
 - Desenvolvedores, analistas, contratantes, operadores

Técnicas de levantamento de requisitos

- Entrevistas
 - Técnica mais comum
 - Analista deve “querer ouvir”
 - Começar com questões pontuais simples e evoluir para mais complexas
 - Tipos de perguntas
 - Abertas: perguntas subjetivas sobre funcionamento de algo
 - Fechadas: perguntas objetivas onde são apresentados valores quantitativos
 - Continuidade: propõe que o entrevistado esclareça sobre um assunto não muito claro

Técnicas de levantamento de requisitos

- Etnografia
 - Observação do trabalho diário
 - Práticas de trabalho são ricas, complexas e dinâmicas
 - Diferença entre o trabalho suposto e o real
- Questionários
 - Sequência lógica de questões a fim de extrair necessidades reais
 - São mais eficazes com *stakeholders* que se expressam melhor por escrito que oralmente
 - O analista deve ter experiência para montar o questionário
 - Pode ser combinado com a entrevista

Técnicas de levantamento de requisitos

- Cenários
 - Descrição de situações reais de trabalho através de exemplos
 - Permite lembrar de detalhes que não podem ser revelados em questionários
 - Podem ser descritos de diversas formas
 - UML ou histórias de usuário
 - Analistas podem simular a execução dos cenários

Qualidade de Requisitos

- A especificação de requisitos deve satisfazer
 - Correção: todo requisito presente é um requisito do software
 - Precisão: todo requisito presente possui única interpretação
 - Completude: reflete todas as decisões de especificações tomadas
 - Consistência: não há conflitos entre subconjuntos de requisitos
 - Priorização: explícita de acordo com a importância e estabilidade
 - Verificabilidade: atestar a conformidade do produto final com os requisitos
 - Modificabilidade: sua estrutura e estilo permitem mudança de forma fácil, completa e consistente
 - Rastreabilidade: permite a fácil identificação dos antecedentes e consequências dos requisitos

Perspective-Based Reading

- Leitura: técnica chave para verificação e validação de documentos
 - Planos de teste, códigos, especificação de requisitos
- PBR: leitura de requisitos escritos em linguagem natural
 - Analisa o documento em busca de certas características que sinalizam problemas
 - Leitura segundo determinado ponto de vista (programador)
 - Ex: ambiguidade número x código
 - Atribui a cada revisor uma perspectiva
 - Cada um recebe um script de aplicação

Diálogo entre cliente e desenvolvedor

- Dificuldades de comunicação merecem atenção especial dos desenvolvedores
- Um dos problemas é a negociação de requisitos entre clientes e desenvolvedores
 - “O cliente não sabe o que quer”
 - É difícil explicar algo antes dele existir
 - “Não pedi porque é óbvio”
 - Usuários ignoram o fato que os desenvolvedores não conhecem seu trabalho
 - “Basta incluir dois campos a mais no formulário”
 - Clientes acham que é fácil fazer modificações no produto
 - “Funcionava mais rápido na fase de testes”
 - Cliente não entende a mudança de ambiente

Ferramentas

- Existem ferramentas que automatizam o levantamento, análise, verificação e documentação de requisitos
- Requisite Pro da IBM/Rational
 - Gerenciamento de mudanças
 - Possibilidade de rastrear requisitos
 - Trabalho cooperativo via web
- Analyst Pro
 - Gerenciamento de requisitos, rastreamento e análise
 - Possível importar em diversos formatos, compartilhá-los e gerenciar mudanças
- Há também gratuitas, como as do site <http://tigris.org>