Slide 2 -Importância

Para entender a importância de um software de qualidade, podemos usar uma história do Therac-25. O Therac-25 era uma máquina de terapia radiológica totalmente controlada por computador. Em 26 de julho de 1985, no Ontário Cancer Foundation no Canadá, um operador, com o paciente dentro da máquina, acionou a máquina e em cinco segundos ela se autodesligou. Como não havia nenhuma mensagem de erro e simplesmente a mensagem “no close” e não a mensagem “treatment paused” o operador verificou o fechamento da máquina novamente e a acionou novamente. O desligamento ocorreu mais cinco vezes e por cinco vezes o operador acionou a máquina. O paciente foi embora sem um resultado de seus exames. Contudo, no dia 3 de novembro do mesmo ano o paciente faleceu. A biópsia revelou que a superexposição à radiação havia causado tantos danos que chegou a afetar de forma tão grave o paciente, que a cabeça do fêmur praticamente foi destruída. Outros casos fatais foram contabilizados e a causa foi isolada e detectada como um defeito de software que possuía procedimentos concorrentes entre si na codificação do software.

Slide 3- por que temos dificuldades em ter qualidade de software.

Slide 4- o que o autor do texto “Cornering the Chimera”, R. Geoffd Romey, diz em relação ao termo qualidade de software:

Vagamente usado em relação ao produto e processo--- de acordo com ele isso tem causado confusão e desviado as empresas de seu objetivo final: melhorar a qualidade dos produtos das fases de desenvolvimento de software. Para ele é importante lembrar que termos como “qualidade” são experimentais. Ou seja, as pessoas julgam dependendo de sua necessidade naquilo, de sua perspectiva. Quais propriedades tangenciais geram uma resposta assim, é diferente. De acordo com Romey, deveríamos foca mais nisso, para melhorarmos a qualidade de software.

Slide 5 - Em relação a “qualidade ser incorporada no software”

Distrai do verdadeiro problema :como fazer software que manifestam atributos de altos níveis—Não podemos fazer atributos de alta qualidade como confiabilidade ou capacidade de manutenção no software, mas sim produzir propriedades de um produto que resultem nisso. E devemos ligar isso a atributos de alta qualidade.

Slide 6- Objetivo de Romey: criar um método para facilitar o que consideramos ser a qualidade de software—de acordo com ele isso facilitaria a implementação de modelos que facilitam e melhoram a qualidade dos softwares. Como dito antes, a qualidade de um software é muito subjetiva e diz respeito a experiência dos usuários , e disso vem a maior dificuldade em solucionar os problemas.

Slide 7 -ROMEY SOBRE ESTUDOS PARA MELHORAR A QUALIDADE DE SOFTWARE: Muitos já foram feitos, mas pararam POR TRÊS MOTIVOS--- de acordo com o autor muitos estudos já foram começados sobre como solucionar o problema da qualidade de software sistematicamente mas ele afirma que esses esforços foram parados devido a 3 principais razões: o tamanho aparente do problema; a grande diversidade de defeitos de qualidade e a dificuldade em transformar esses atributos em propriedades mais tangíveis.

Slide 8 - A sugestão proposta pelo autor é criar um modelo genérico de qualidade e um processo para dar a esse modelo as especificidades necessárias para funcionar nos diferentes sistemas, os principais elementos desse modelo seriam a escolha de componente, as propriedades do componente e a composição do componente. Cada componente possui suas *rules-of-form* (regras de forma) e *rules-of-composition* (regras de composição) que não podem ser violadas, caso o sejam, dependendo da gravidade da violação, o software pode ter perdas de funcionalidade.

Slide 9 - Classificação de categorias: propriedades de correção; propriedades internas; propriedades contextuais; e propriedades descritivas---- As melhores categorias de propriedade de software segundo Dromey se classificam em: propriedades de correção; propriedades internas; propriedades contextuais; e propriedades descritivas. Depois de identificar todas as propriedades nos resta ligar as propriedades tangíveis aos atributos intangíveis, trabalho que o autor afirma que infelizmente não é viável de ser feito para todas as propriedades então propõe fazermos somente para as 4 supracitadas.

Slide 10 – Propriedades de Correção: As coisas podem dar errado ou com a forma como os componentes são implantados diretamente ou em contexto. Algumas propriedades são tão significativas que se são violadas, o produto não executará como pretendido. Algumas propriedades de correção merecem uma consideração separada então são classificadas separadamente. As propriedades de correção podem ser internas - associado a componentes individuais - ou contextuais– associado com a forma como os componentes são usados

contexto.

Slide 11- Propriedades internas: Todo componente tem uma forma normal que define sua “verdadeira” estrutura interna. O corpo de um loop por exemplo, deve sempre progredir em direção a finalização. A forma normal de um componente não deve ser violada, independentemente do contexto. Propriedades internas medem quão bem um componente foi implementado de acordo com sua intenção de uso ou implementação pretendida ou quão bem foi composta.

Slide 12- Propriedades contextuais: Lidam com influências externas por e sobre o uso de um componente. Essa abordagem simplifica signitivamente a tarefa de lidar com problemas de qualidade que provém de grandes números ou diferentes componentes. Às vezes envolvem correção, mas muitas vezes envolvem problemas menos severos que afetam atributos de qualidade de alto nível como a capacidade de manutenção.

Slide 13-Propriedades Descritivas: para ser útil, um software precisa ser fácil de entender e atender ao seu uso proposto. As propriedades descritivas se aplicam para requisitos, designs, implementação e interface do usuário. Por exemplo, variáveis de um programa devem ter nomes descritivos. Às vezes, propriedades descritivas tem o importante papel de especificar a funcionalidade e restrições, ou identificar contextos, formalmente ou informalmente.

Slide 14-Modelo do autor: primeiro são identificados atributos de qualidade para o produto, em seguida identificar os componentes do produto, identificar e classificar as principais propriedades de qualidade, ligar as propriedades a qualidade de atributos e por fim avaliar o modelo, verificar suas falhas e ou refina-las ou recomeçar o processo com diferentes variáveis.

Slide 15- só dizer o que tá escrito e explicar a imagem

Silde 16- Referências- a primeira é do artigo e a segunda de onde a história inicial foi retirada.