

Aula 6

Gerência de configuração e evolução

Profª Adriana Bastos da Costa

1

Conversa Inicial

2

Gerência de configuração

- A gerência de configuração é importante para manter todas as mudanças e os componentes criados para satisfazê-las organizados e gerenciados
- A gerência de configuração continua sendo a maior aliada da gestão correta e completa dos componentes de software, enquanto eles forem criados e evoluídos

3

Evoluções do software

- São muitas as implicações para o software, trazidas pelas modificações ocorridas ao longo do tempo
- Nessa aula, vamos discutir sobre alguns temas que envolvem os riscos e as necessidades de garantir que o software se mantenha íntegro, mesmo quando passa por constantes alterações

4

- Esta aula está dividida em 5 temas principais, sendo eles:
 - Processo de evolução
 - Sistemas legados
 - Riscos de substituição do software
 - Reengenharia de software
 - Refatoração

5

Processo de evolução

6

Processo de evolução do software

- De acordo com Sommerville (2019), existe uma necessidade constante e inevitável de evolução por parte de todo software que espera manter-se ativo por um longo período de tempo

7

Processo de evolução do software

- As alterações são naturais e fazem parte de manter um software útil para seus usuários
- Isso acontece porque os sistemas refletem situações do mundo real, logo há a necessidade de que o software mude, acompanhando as mudanças de requisitos impostos pelo ambiente em que está inserido

8

Processo de evolução do software

- É preciso compreender o requisito que será incluído ou alterado no software, é preciso minimizar o risco de manter um software que está em uso por vários usuários e que pode também impactar outros softwares que estão em uso e que fazem interface com o software que está em manutenção

9

Processo de evolução do software

- O processo de manutenção do software é uma organização que funciona como uma engrenagem e que precisa ser analisada e gerenciada como um todo, para garantir que o negócio continue estável e em funcionamento adequado, gerando os resultados esperados

10

Sistemas legados

11

Sistemas legados

- Um problema-chave para as organizações é implementar e gerenciar a manutenção em seus sistemas legados
- Sistemas legados são sistemas imprescindíveis para os negócios de uma organização

12

Sistemas legados

- Em geral, possuem documentação precária ou inexistente e passaram, ao longo dos anos, por manutenções realizadas por diversos profissionais, muitas vezes sem seguir boas práticas de engenharia de software

13

Sistemas legados

- Os softwares em funcionamento em uma empresa são chamados de software legado
- Um sistema legado é um desafio na rotina das empresas, pois ele é imprescindível para que os processos funcionem, mas, por outro lado, podem estar passando por um processo natural de envelhecimento

14

Sistemas legados

- Recursos lentos, *bugs*, ausência de funcionalidades, travamento e perda de dados são algumas das dificuldades enfrentadas e que afetam diretamente o resultado do trabalho dos colaboradores que utilizam sistemas legados

15

Sistema legado - problemas

- Software desatualizado - é sempre ruim ter um software desatualizado, pois, quando um sistema deixa de passar por atualizações, ele não só está sujeito a apresentar *bugs*, como também a ameaças virtuais, a exemplo de vírus

16

Sistema legado - problemas

- Falta de mobilidade - em casos em que não há mobilidade, quando um colaborador externo precisar acessar um relatório da empresa, algo bastante simples, isso se torna impossível, pois os dados só são acessíveis por meio de um computador. Com o crescimento do uso de aplicativos móveis, passa a ser essencial ter uma versão *mobile* dos softwares

17

Sistema legado - problemas

- Falta de escalabilidade - a escalabilidade é uma característica essencial em um software que tende a crescer e a ter mais usuários com o passar do tempo. Geralmente os sistemas legados não foram desenvolvidos a partir dessa perspectiva, o que se torna um desafio para mantê-los estáveis e disponíveis para uso quando passa da sua capacidade projetada

18

Sistema legado - problemas

- **Falta de suporte** - no caso de sistemas legados, pode haver o problema da falta de suporte. Um sistema que recentemente deixou de ter suporte foi o Windows XP, por exemplo. Para poder usar versões mais recentes dos programas da Microsoft, é preciso que o sistema operacional seja atualizado

19

Sistema legado - problemas

- **Incompatibilidade com sistemas modernos** - a integração com sistemas modernos, muitas vezes, é limitada no caso dos legados, indo totalmente contra a lógica de softwares, que devem ser sempre compatíveis com outras ferramentas. Isso pode prejudicar e impactar fortemente um negócio que busca ser mais ágil e eficiente, principalmente em relação aos seus clientes

20

Sistema legado - problemas

- **Software hospedado em servidor físico** - a maioria dos softwares que têm grande dependência dos dispositivos de hardware podem se configurar como sistemas legados. A hospedagem em servidores físicos tem um valor muito mais alto que o custo de manter os softwares na nuvem

21

Riscos de substituição do software

22

Manutenção de software

- **Apesar das dificuldades de gerenciar a manutenção de um software**, é fundamental manter os softwares atualizados. E, muitas vezes, gestores e proprietários de empresas não sabem da importância da atualização dos softwares, sejam estes utilizados no cotidiano dos processos corporativos internos ou no atendimento ao cliente

23

Risco para o negócio

- **Um software desatualizado pode trazer riscos para o bom andamento dos negócios**, podendo causar prejuízos gravíssimos que poderiam ser evitados se a manutenção adequada dos sistemas fosse executada e gerenciada de maneira padronizada pelos responsáveis

24

O grande conflito

- Se a empresa opta pela inclusão ou atualização de novas tecnologias, a segurança dos dados pode ficar comprometida, pois, durante o processo de modernização, as informações correm o risco de serem perdidas, corrompidas ou até mesmo roubadas
- Se a empresa opta por não atualizar e manter seus softwares, ela pode estar perdendo competitividade e, consequentemente, perdendo dinheiro

25

Processo de manutenção de software

- Por isso, é importante definir um processo completo e robusto de construção e manutenção de software, de forma a gerenciar os riscos envolvidos e inserir controles que diminuam os possíveis impactos nos negócios das evoluções de um software

26

Processo de manutenção de software

- O processo de manutenção de software precisa ser organizado para que o projeto seja realizado em partes, trazendo melhoria para a operação mais rapidamente
- A finalização de uma parte do processo de modernização, considerada pequena em relação ao todo, já é capaz de trazer resultados visíveis e de valor para a empresa

27

Processo de manutenção de software

- É preciso trabalhar para que todo o projeto seja concluído e forma priorizada. Ter acesso a melhorias antes do término da modernização, e contentar-se com elas, pode desestimular a equipe a seguir com o projeto, e a implementação do novo sistema nunca ser dada como pronta, o que não entregará todo o valor esperado pela empresa

28

Processo de manutenção de software

- É fundamental ter uma política eficiente de *backup* e de recuperação de dados, que possa ser usada em caso de emergência. A contingência de ter uma política bem estruturada de *backup* pode ser cara e necessitar de investimentos, mas garante a tranquilidade de se manter os negócios funcionando por mais tempo

29

Reengenharia de software

30

Reengenharia

- O conceito não é exclusivo de software; ele surgiu na década de 90 como uma resposta ao turbulento mundo que exigia ações rápidas e constantes para garantir a competitividade
- “Reengenharia” significa abandonar antigos conceitos e analisar em detalhes todo trabalho necessário para produzir os produtos ou serviços

31

Reengenharia de software

- É o processo de reconstrução de um software já existente e em uso
- Engenharia reversa é o processo de descobrir os princípios tecnológicos e o funcionamento de um software, através da análise de sua estrutura, função e operação

32

Reengenharia X Engenharia reversa

- Quando falamos em hardware, a engenharia reversa desmonta algo para conhecer seu funcionamento, enquanto que a reengenharia analisa algo para conhecer seu funcionamento
- Mas, em software, os conceitos realmente são muito próximos

33

Reengenharia de software

- Com a rápida evolução da tecnologia, as empresas acabam ficando com três possíveis alternativas para se manter competitivas, sendo elas:

34

Reengenharia de software

- Manter os softwares legados, afetando a produtividade dos colaboradores e com custos cada vez maiores
- Reconstruir os softwares novamente
- Realizar a reengenharia para aumentar a manutenibilidade ou para implementar os softwares em um paradigma mais atual

35

Reengenharia de software

- Não é uma decisão fácil de ser tomada, pois envolve analisar o custo e o benefício esperado com cada uma das alternativas

36

Reengenharia de software

- Segundo Sommerville (2007), a principal característica da reengenharia de software é o risco reduzido, ou seja, a chance de ocorrer problemas ou erros durante o redesevolvimento de um software é maior do que o da reengenharia. Sobre os custos reduzidos, ele afirma que o custo da reengenharia é em torno de quatro vezes menor do que o utilizado para desenvolver um software novo

37

Refatoração do software

38

Refatoração

- Baseado nos conceitos do XP, a refatoração é sinônimo de código limpo
- Um código limpo é um código simples, elegante e direto, fácil de ser mantido
- Isso faz todo sentido quando falamos que um software sofre, inevitavelmente, constantes manutenções ao longo da sua vida

39

Refatoração

- É uma forma diferente de pensar o desenvolvimento do software, é uma mudança cultural
- Nem todos os desenvolvedores fazem refatoração, pois acreditam que é um retrabalho que não vai contribuir com o seu dia a dia, ou, ainda, porque não aprenderam a importância desse hábito

40

Refatoração

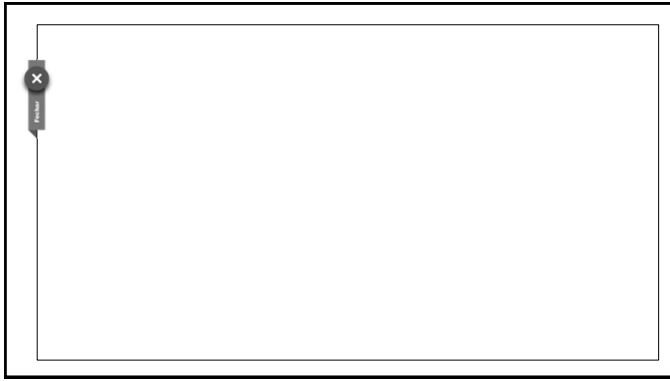
- É preciso que todos entendam que, em um código mal escrito, ao implantar uma nova funcionalidade, é muito provável que os desenvolvedores acabem perdendo ainda mais tempo no ajuste do código

41

Refatoração - benefícios

- Melhora a qualidade do código, torna-o mais legível e fácil de entender
- Previne defeitos, já que é possível encontrar falhas no desenvolvimento de pequenos trechos do código
- Aumenta a produtividade, pois essa "limpeza" fará a equipe de desenvolvimento programar mais rápido, já que fica mais fácil incluir uma nova funcionalidade

42



43