

SOFTWARE ARCHITECTURE

ARQUITETURA DE SOFTWARE

AULA 03

SUMÁRIO

O QUE VEM POR AÍ?	3
HANDS ON	4
SAIBA MAIS	5
O QUE VOCÊ VIU NESTA AULA?	12
REFERÊNCIAS	13
PALAVRAS-CHAVE	14

O QUE VEM POR AÍ?

Nesta aula você vai aprender alguns dos pilares de uma boa arquitetura de software. Iniciaremos falando sobre documentação, depois avançaremos nos testes e, por fim, vamos aprender sobre modificabilidade, que é muito importante conhecer.

EMAN

HANDS ON

No primeiro vídeo desta aula, os professores apresentaram alguns tipos de documentação de softwares e como esses documentos podem ajudar no processo de levantamento de requisitos, até a parte de manutenção e desenvolvimento de novas funcionalidades do sistema.

Na segunda aula em vídeo, os docentes fizeram um handsOn para apresentar o Swagger e o Redoc, ferramentas importantes para a criação destas documentações.

Na parte de testes, os professores apresentaram o teste do “caminho feliz” dentro de um sistema legado, e o teste correto utilizando o BDD, apresentado na aula anterior.

No vídeo final desta aula, os professores abordaram o tema de modificabilidade demonstrando, por meio de exemplos práticos, como é difícil atualizar um sistema legado que foi construído sem muitos requisitos, e como é simples quando começamos o desenvolvimento por meio dos pontos que vimos nos vídeos anteriores.

Assista aos vídeos para aprender muito com os nossos experts!

SAIBA MAIS

DOCUMENTAÇÃO, TESTABILIDADE E MODIFICABILIDADE

Documentação, testabilidade e modificabilidade são fatores importantes que influenciam a qualidade do software produzido.

Esses são passos simples que, se adotados desde o início da concepção do software, são fáceis de se manter ao longo do desenvolvimento.

Para que estes pontos fiquem bem claros para você, vamos detalhar cada um deles por meio de exemplos teóricos e práticos.

Começando pela documentação de software, existem muitos tipos de documentações, como:

- Documentação de utilização do projeto;
- Documentação de desenvolvimento do projeto;
- Documentação da infraestrutura do projeto;
- Documentação da arquitetura do projeto... etc.

Cada tipo de documentação tem sua própria importância e função, e todas elas juntas ajudam a garantir que o software seja compreendido e mantido ao longo do tempo.

A documentação de utilização do projeto, por exemplo, ajuda os usuários finais a entender como usar o software, enquanto a documentação de desenvolvimento do projeto pode ser usada por outros desenvolvedores para entender como o software foi construído. A documentação da infraestrutura do projeto é útil para garantir que a configuração do ambiente de desenvolvimento e produção seja consistente e adequada, enquanto a documentação da arquitetura do projeto ajuda a garantir que o software seja escalável e possa ser expandido conforme necessário.

Agora que conhecemos um pouco de cada uma delas, que tal alguns exemplos?

DOCUMENTAÇÃO DE UTILIZAÇÃO DO PROJETO

Um exemplo de documentação de utilização do projeto pode ser um manual do usuário que explique como usar o software, quais são as suas funcionalidades e como realizar tarefas específicas dentro dele.

Esse tipo de documentação pode ser útil tanto para usuários finais e para outros membros da equipe de desenvolvimento que precisam se familiarizar com o software.

A documentação de utilização do projeto geralmente inclui instruções passo a passo, imagens, vídeos, fluxogramas e outras formas de ajudar os usuários a entender e usar o software de forma eficaz.

DOCUMENTAÇÃO DE DESENVOLVIMENTO DO PROJETO

A documentação de desenvolvimento do projeto é voltada para o time de desenvolvimento do software, e tem como objetivo registrar os processos de desenvolvimento, desde a concepção até a entrega do software. Essa documentação pode incluir informações sobre as tecnologias utilizadas, decisões de arquitetura e design, testes realizados, bugs encontrados, soluções adotadas, entre outras coisas.

Um exemplo de documentação de desenvolvimento é um documento que descreve a arquitetura do sistema, detalhando cada camada, seus componentes e como eles se comunicam.

Esse documento pode incluir diagramas de arquitetura, fluxogramas e especificações técnicas. Outro exemplo é um relatório de testes, que descreve os testes realizados no software e seus resultados, incluindo os testes unitários, de integração, funcionais e de desempenho.

DOCUMENTAÇÃO DA INFRAESTRUTURA

A documentação da infraestrutura é um conjunto de documentos que descrevem os componentes e configurações da infraestrutura de TI de uma organização. O objetivo é fornecer informações detalhadas sobre os componentes de hardware, software e rede da infraestrutura, para facilitar a manutenção e o gerenciamento.

Um exemplo de documentação da infraestrutura pode incluir:

- Inventário de hardware: uma lista detalhada de todos os dispositivos de hardware em uso na infraestrutura, incluindo servidores, roteadores, switches, firewalls, dispositivos de armazenamento etc. Para cada dispositivo, a documentação deve incluir informações como modelo, número de série, localização física, configuração de rede, configurações de BIOS/firmware e histórico de manutenção.
- Inventário de software: uma lista detalhada de todo o software em uso na infraestrutura, incluindo sistemas operacionais, aplicativos, utilitários, ferramentas de monitoramento etc. Para cada software, a documentação deve incluir informações como nome, versão, data de instalação, configurações de segurança e licenças.
- Diagramas de rede: um conjunto de diagramas que mostram como os componentes da infraestrutura de TI se conectam e interagem. Isso pode incluir diagramas de topologia de rede, diagramas de conectividade lógica e diagramas de fluxo de dados.
- Políticas e procedimentos: documentação que descreve as políticas e procedimentos de segurança, manutenção e uso da infraestrutura de TI. Isso pode incluir políticas de backup, políticas de acesso remoto, procedimentos de manutenção preventiva e procedimentos de recuperação de desastres.
- Documentação de configuração: documentação que descreve as configurações específicas de cada componente da infraestrutura de TI. Isso pode incluir informações como endereços IP, configurações de firewall, configurações de segurança e configurações de armazenamento.
- Relatórios de monitoramento: relatórios que mostram o desempenho da infraestrutura de TI ao longo do tempo. Isso pode incluir relatórios de uso de recursos, relatórios de falhas e relatórios de alerta.

DOCUMENTAÇÃO DA ARQUITETURA DO PROJETO

A documentação da arquitetura do projeto é um conjunto de documentos que descrevem a estrutura e o design do projeto de software. O objetivo é fornecer informações detalhadas sobre a arquitetura, padrões de design, componentes e interfaces do sistema, para facilitar o desenvolvimento, manutenção e evolução do projeto.

Um exemplo de documentação da arquitetura do projeto pode incluir:

- Visão geral da arquitetura: uma visão geral da arquitetura do sistema, incluindo os objetivos de design, requisitos, restrições e decisões de design importantes.
- Diagramas de arquitetura: diagramas que mostram a estrutura do sistema, incluindo componentes, interfaces, dependências e fluxos de dados. Isso pode incluir diagramas de blocos, diagramas de sequência, diagramas de estado e outros tipos de diagramas.
- Padrões de design: uma descrição dos padrões de design usados no sistema, incluindo padrões de arquitetura, padrões de componentes, padrões de comportamento e outros padrões relevantes. Isso pode incluir uma lista de padrões com exemplos de onde são usados no sistema.
- Descrição dos componentes: uma descrição detalhada dos componentes do sistema, incluindo seus objetivos, funcionalidades e interfaces. Isso pode incluir descrições de cada componente, incluindo como ele se comunica com outros componentes e o que ele faz.
- Interfaces do Sistema: uma descrição das interfaces do sistema, incluindo APIs, protocolos, formatos de dados e outras interfaces relevantes. Isso pode incluir uma lista de interfaces, com exemplos de como são usadas no sistema.
- Decisões de design: uma descrição das principais decisões de design tomadas durante o desenvolvimento do projeto, incluindo os motivos por trás dessas decisões. Isso pode incluir decisões sobre arquitetura, escolha de tecnologia, design de interfaces e outros aspectos importantes do projeto.

TESTABILIDADE DE SOFTWARE

Testabilidade de software é a capacidade de um sistema de software de ser testado de maneira eficaz. Um software é considerado testável quando pode ser facilmente avaliado em relação à sua qualidade, funcionalidade e desempenho. Isso inclui a facilidade com que testes podem ser criados, executados e mantidos.

A testabilidade de software é importante porque é difícil garantir que um sistema de software atenda aos requisitos sem testá-lo adequadamente. Além disso, quanto mais testável o software for, menos tempo e recursos serão necessários para testá-lo, o que pode ajudar a reduzir o custo e o tempo de desenvolvimento.

Alguns fatores que afetam a testabilidade de software incluem:

- Design de software: um bom design de software pode tornar o código mais modular, legível e reutilizável, facilitando a criação de testes.
- Arquitetura de software: a arquitetura do sistema pode afetar a facilidade de criação de testes. Uma arquitetura bem projetada pode reduzir a complexidade do sistema e torná-lo mais fácil de testar.
- Instrumentação de código: a instrumentação de código envolve a adição de código adicional para coletar dados para análise, monitoramento e teste. Isso pode ajudar a identificar problemas no software e melhorar a sua testabilidade.
- Dados de teste: a disponibilidade de dados de teste relevantes e realistas é essencial para criar testes eficazes.
- Ferramentas de teste: o uso de ferramentas de teste automatizadas pode ajudar a aumentar a eficiência e eficácia dos testes.
- Acessibilidade de interfaces: interfaces bem definidas e documentadas podem facilitar a criação de testes.

Ao projetar e desenvolver um software, é importante levar em consideração a testabilidade do mesmo, no intuito de garantir a qualidade e funcionalidade do produto final. A testabilidade deve ser considerada como um requisito importante do software, e não como um processo secundário ou opcional.

MODIFICABILIDADE

Modificabilidade de software é a capacidade de um sistema de software de ser modificado facilmente. Isso inclui a facilidade com que o software pode ser alterado, mantido e evoluído para atender a novos requisitos e necessidades.

A modificabilidade de software é importante porque os sistemas de software precisam ser modificados com frequência para corrigir defeitos, melhorar a funcionalidade, lidar com novos requisitos ou adaptar-se a novos ambientes. A capacidade de modificar o software de forma eficiente e eficaz pode reduzir o tempo e o custo de desenvolvimento, melhorar a qualidade do software e garantir que o sistema atenda às necessidades do usuário.

Alguns fatores que afetam a modificabilidade de software incluem:

- Design de software: um design de software modular e bem estruturado pode facilitar a modificação do software. Os componentes do software devem ser claramente definidos e as interações entre esses componentes devem ser bem documentadas.
- Arquitetura de software: a arquitetura do sistema pode afetar a facilidade de modificação. Uma arquitetura bem projetada pode facilitar a adição ou remoção de componentes bem como a evolução do sistema.
- Qualidade do código: o código de software deve ser fácil de entender e manter. Isso inclui a legibilidade do código, a utilização de padrões de codificação, a minimização de código duplicado e a utilização de nomes significativos de variáveis e funções.
- Documentação: a documentação do software deve ser completa e atualizada. Isso inclui documentação de design, documentação de código e documentação de processo.
- Testabilidade: a testabilidade do software é importante para facilitar a modificação do software. O software deve ser testável e ter testes automatizados para garantir que as alterações feitas no software não causem regressão ou defeitos.

Ao projetar e desenvolver um software, é importante levar em consideração a modificabilidade do mesmo, a fim de garantir que o sistema possa ser facilmente adaptado às mudanças no ambiente e às necessidades do usuário. A modificabilidade deve ser considerada como um requisito importante do software.

EMENDAS

O QUE VOCÊ VIU NESTA AULA?

Nesta aula, nós aprendemos três pontos importantes que devemos levar em consideração no início do projeto até as suas modificações futuras. Estes pontos facilitam a implementação de uma nova funcionalidade até a manutenção nos módulos já existentes.

Depois de uma aula fenomenal como esta, que tal conversar com os seus colegas sobre o conteúdo na nossa comunidade no Discord? Participe para aproveitar ainda mais essa experiência Pós-Tech!

REFERÊNCIAS

ANICHE, M. **Test-driven Development. Teste e Design no Mundo Real com .net.** São Paulo: Casa do código, 2013.

_____. **Testes Automatizados de Software. Um Guia Prático.** São Paulo: Casa do código, 2015.

VAZQUEZ, C. E.; SIMÕES, G. S. **Engenharia de requisitos: software orientado ao negócio.** Rio de Janeiro: Brasport, 2016

EMANIP

PALAVRAS-CHAVE

Arquitetura de software. Modularização. Processo de Desenvolvimento de Software.

EXEMPLO



POSTECH