METODOLOGIA ÁGIL: BDD (BEHAVIOR DRIVEN DEVELOPMENT)

Patrick Waschburger, Diego Correa, Kevin Soares e Fabricio Rech

Resumo: *Usando linguagem natural para definir os comportamentos de software desejados, o Behavior Driven Development (BDD) é uma técnica ágil que coloca forte ênfase na comunicação eficaz e no trabalho em equipe de desenvolvimento. O BDD incentiva o desenvolvimento de software de alto calibre que esteja alinhado às necessidades do usuário, promovendo os conceitos de feedback imediato, comunicação aberta e trabalho em equipe. Seus métodos para desenvolver cenários de teste e histórias de usuários que apoiam a precisão e eficácia do desenvolvimento de software.*

Abstract: *Using natural language to define desired software behaviors, Behavior Driven Development (BDD) is an agile technique that places a strong emphasis on effective communication and development teamwork. BDD encourages the development of high-caliber software that is aligned with user needs, promoting the concepts of immediate feedback, open communication, and teamwork. Its methods for developing test designs and user stories that support the accuracy and effectiveness of software development.*

Introdução

Behavior Driven Development (BDD), ou traduzindo, Desenvolvimento Guiado por Comportamento, é uma metodologia ágil que utiliza principalmente a colaboração entre todos participantes do projeto, principalmente pela utilização da linguagem natural para descrever os comportamentos que o software deve realizar. Assim, o BDD possibilita que haja uma comunicação clara entre todos da equipe, ajudando a identificar erros e corrigi-los. Quando é ajustado um teste automatizado, o BDD garante que o código realize todos os critérios definidos pelos cenários, resultando em um software de qualidade e que atenda as necessidades dos usuários

Princípios fundamentais

Os pilares do BDD são essenciais para o seu sucesso. Em primeiro lugar, o trabalho em equipe é um princípio essencial do BDD. Ele garante que todos da equipe estejam cientes dos requisitos, promovendo a colaboração entre eles.

Além disso, a comunicação clara é outro princípio fundamental do BDD. Uma linguagem natural, frequentemente no estilo Dado-Quando-Então, torna possível explicar os comportamentos pretendidos do sistema de uma maneira clara e compreensível. Este princípio

deixa mais fácil para todas as partes se entenderem, assim diminuindo os risco de falhas de comunicação e garantindo que todos tenham a mesma ideia para o projeto.

O BDD também destaca o feedback imediato, fazendo com que os testes automatizados facilitem o recebimento do feedback rápido sobre o sistema. Isso permite identificar problemas no início do ciclo de desenvolvimento e corrigi-los para produzir um produto de qualidade e que atenda as necessidades exigidas.

Componentes

- **Histórias de Usuário**

As histórias de usuário são pequenas descrições das funcionalidades do sistema, escritas do ponto de vista do usuário. Elas absorvem os requisitos de forma clara e acessível, mostrando o valor de cada funcionalidade para o usuário.

O formato normalmente utilizado é “Como, eu quero, para que”, sendo um exemplo: Como dono do mercado, eu preciso comprar produtos, para que eu possa vender em meu mercado mais tarde.

Essa forma simples ajuda toda a equipe a entender o que é necessário e por que é importante, facilitando a priorização e o desenvolvimento das funcionalidades.

- **Cenários de Teste**

Os cenários de teste descrevem em detalhes como uma funcionalidade deve se comportar em diferentes situações. Eles são escritos de uma maneira que todos possam entender, sendo utilizado linguagem natural.

O formato normalmente utilizado é “Given (Dado), When (Quando),Then (Então)”, tendo como exemplo: Dado que o carrinho está vazio, quando eu adiciono um produto ao carrinho, então o carrinho deve conter um produto.

Os cenários de teste ajudam a garantir que todos entendam o comportamento esperado do sistema e podem ser usados tanto para testes manuais quanto automatizados.

- **Ferramentas**

Existem várias ferramentas que suportam o BDD, facilitando a escrita, execução, entre outros. Algumas das mais populares são:

-Cucumber: Uma das ferramentas mais conhecidas para BDD, permite escrever testes automatizados em uma linguagem simples e compreensível.

-SpecFlow: Uma ferramenta que tem como objetivo traduzir as histórias de usuários para a linguagem de programação.
-Behave: Uma ferramenta BDD para Python, com uma sintaxe clara e simples para escrever cenários de teste.

Essas ferramentas transformam os cenários de teste em scripts automatizados, garantindo que o comportamento esperado do sistema seja verificado continuamente durante o desenvolvimento.

Benefícios

O BDD melhora a comunicação, reduz erros e facilita a documentação em tempo real. Um dos maiores benefícios do BDD é a melhoria da comunicação porque incentiva uma comunicação clara entre os membros da equipe e os clientes. O BDD reduz significativamente a probabilidade de mal-entendidos, garantindo que todos os envolvidos tenham uma compreensão clara e compartilhada dos comportamentos esperados do sistema. Isso é feito por meio do uso de uma linguagem natural de fácil compreensão tanto para técnicos quanto para não técnicos.

O BDD também funciona para reduzir erros, permitindo a detecção e correção de problemas, criando um sistema mais forte e confiável ao especificar explicitamente os comportamentos esperados e automatizar os testes. As taxas de erro são reduzidas significativamente com este método ao longo do ciclo de desenvolvimento.

A documentação viva é outro grande benefício. O BDD permite a criação de documentação dinâmica e atualizada do sistema, portanto esta opção não é necessária. Os testes comportamentais servem como documentação viva, oferecendo mais dinamismo na criação de documentos relacionados ao sistema, o que facilita a manutenção e evolução do software.

Desafios

A curva de aprendizado, a especificação de cenários e a manutenção de testes são aspectos cruciais para a prática eficaz do BDD.

- **Curva de aprendizado**

O BDD tem poucos requisitos para sua execução, porém tendo um potencial para melhora muito grande, fazendo sua curva de aprendizado ser simples para fazer o básico, mas com o tempo e a prática, uma melhora constante provavelmente será perceptível.

A base de conhecimento necessária para a prática do BDD é simples por comumente ser usado o gherkin, que com pouca prática já se pega o jeito, assim como ferramentas como cucumber, SpecFlow, Behat, entre outros, que facilitam o processo, mesmo para pessoas que não conhecem nenhuma linguagem de programação.

A parte mais importante e mais difícil do aprendizado usando BDD é principalmente na mentalidade, focando no comportamento assim como na colaboração da equipe. Como o BDD é enfatizado no comportamento do software em relação aos exemplos concretos, pode ser uma grande mudança para equipes acostumadas com outros testes, assim como terão de se adaptar à colaboração com mais pessoas e mais constante do que quando se utilizam outros métodos.

- **Especificação de cenários**

A especificação de cenários usando o BDD é talvez o passo mais importante para aprendê-lo de maneira adequada, e para fazê-la do jeito correto é importante focar em dois aspectos

- A clareza e o detalhamento de todos os casos é um aspecto que se deve focar, pois a linguagem natural que é usada no BDD pode dar espaço para muita ambiguidade e diferentes interpretações, então ser claro e específico em todos os cenários é uma prioridade.

Cobrir a maior quantidade possível de casos, explorando a maioria das possiblidades dentro do software, é comum ignorar casos mais raros e específicos, o que é um hábito que deve se evitar ao máximo. Outra parte importante é a priorização dos casos, escolher quais casos devem ser escritos primeiro e quais se podem adiar é um fator essencial ao escrever casos.

- **Manutenção de testes**

Manter os testes atualizados é uma parte que mesmo que seja esquecida frequentemente, também tem um grande valor quando se fala em BDD, manter os testes sincronizados com as mudanças nos requisitos pode ser bem difícil e trabalhoso, então atualizar os cenários conforme o software evolui ajuda a manter a eficiência do BDD. Os testes precisam sempre se manter atualizados também, já que com o tempo eles podem se tornar obsoletos, fazendo com que uma refatoração seja necessária.

Evitar cenários muito longos e complexos, pois eles se tornam muito complexos de se manter e entender, manter os cenários simples também é essencial para o BDD se manter funcionando de maneira ideal. Também é bom manter os cenários independentes um dos outros, cenários que dependem de outros cenários são frágeis e difíceis de gerenciar, é bom evitar sempre que possível.

Conclusão

O Behavior-Driven Development (BDD) é uma metodologia ágil que promove uma comunicação clara dentro da equipe, descrevendo os comportamentos desejados do software por meio de colaboração e linguagem natural. Três de seus princípios principais são feedback imediato, comunicação transparente e trabalho em equipe. As duas partes principais do BDD são cenários de teste, que descrevem o comportamento esperado do sistema em vários contextos, e histórias de usuários, que encapsulam requisitos de forma concisa. SpecFlow, Behave e Cucumber são algumas das ferramentas que facilitam a prática do BDD, convertendo cenários de teste em scripts automatizados.

As vantagens do Desenvolvimento Orientado a Comportamento (BDD) abrangem comunicação aprimorada, menos erros e documentação dinâmica; no entanto, as desvantagens incluem uma curva de aprendizado acentuada, desenvolvimento meticuloso de cenários e manutenção contínua de testes. Concluindo, o BDD é um método bem-sucedido para criar software de alta qualidade que atenda às necessidades do usuário de forma rápida e eficaz, enfatizando o trabalho em equipe, a clareza e o feedback contínuo.

Referências

OBJECTIVE SOLUÇÕES EM TI. BDD: Behavior Driven Development. Disponível em: https://www.objective.com.br/insights/bdd/. Acesso em: 03 jul. 2024.

O QUE É BDD (Behavior Driven Development) e como funciona? YouTube, 2022. Disponível em: https://youtu.be/KahnYIs7mYI?si=u45KoiuGnkSe_77n. Acesso em: 04 jul. 2024.

BDD: Behavior Driven Development na prática com Cucumber. YouTube, 2022. Disponível em: https://youtu.be/HH-m46ldctw?si=DRFI8iFHglMwf7Qq. Acesso em: 02 jul. 2024.

Como aplicar BDD (Behavior Driven Development) no seu projeto. YouTube, 2022. Disponível em: https://youtu.be/DoEzwnqB1mk?si=d79N3CeF-BCDcibJ. Acesso em: 03 jul. 2024.

Vantagens do BDD (Behavior Driven Development) na qualidade do software. YouTube, 2022. Disponível em: https://youtu.be/BDKmhma2mN0?si=22EEa9wrX5xZZaOz. Acesso em: 04 jul. 2024.

Implementando BDD: Exemplos práticos e melhores práticas. YouTube, 2022. Disponível em: https://youtu.be/gXh0iUt4TXA?si=DCM7hYMVrB0zbXGd. Acesso em: 04 jul. 2024.

COODESH. BDD na prática: entenda o que é e como funciona. 2023. Disponível em:https://coodesh.com/blog/candidates/metodologias/bdd-na-pratica-entenda-o-que-e-e-como-funciona/. Acesso em: 04 jul. 2024.

LINKEDIN. How do you ensure quality and maintainability? Disponível em: https://pt.linkedin.com/advice/0/how-do-you-ensure-quality-maintainability-1e?lang=pt. Acesso em: 01 jul. 2024.

DIO. Escrita de cenário de teste usando BDD. 2023. Disponível em: https://www.dio.me/articles/escrita-de-cenario-de-teste-usando-bdd. Acesso em: 01 jul. 2024.

ANDERLE, Angelita. Ferramentas de Behavior Driven Development (BDD): uma revisão na literatura cinzenta. Disponível em: http://repositorio.jesuita.org.br/bitstream/handle/UNISINOS/5314/Angelita+Anderle-Monografia_.pdf?sequence=1. Acesso em: 04 jul. 2024.

Ferramentas de Behavior Driven Development (BDD): uma revisão na literatura cinzenta. Disponível em: https://scholar.archive.org/work/adbathxjqfhohh7pct3qvv75ji/access/wayback/https://s3-eu-west-1.amazonaws.com/pfigshare-u-files/31103740/FerramentasdeBehaviorDrivenDevelopmentBDDUmaRevisaonaLiteraturaCinzenta.pdf?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Expires=10&X-Amz-SignedHeaders=host&X-Amz-

Signature=6480bb4bcf48ebc85c7afaf2ac5b6ac73f02208561f9d5342143bc0ba8276948&X-Amz-Date=20211020T115900Z&X-Amz-Credential=AKIAIYCQYOYV5JSSROOA/20211020/eu-west-1/s3/aws4_request. Acesso em: 04 jul. 2024.