UNIVERSIDADE DA REGIÃO DE JOINVILLE - SC
http://www.univille.edu.br/
Disciplina: Engenharia de Requisitos de Software
Professor:GLAUCIO SCHEIBEL (glaucio.scheibel@univille.br)
Aluno:Alessandro Dos Santos Marques (alessandromarques@univille.br)

Divida técnica

1. Defina divida técnica.

   R: Descreve a dívida que a equipe de desenvolvimento assume quando escolhe um design ou abordagem fácil de implementar no curto prazo mas com grande impacto negativo no longo prazo.

   A dívida técnica é similar à dívida financeira, onde assume- se um compromisso de entregar um projeto com algumas deficiências que, geralmente estéticas, em razão de dar celeridade a criação do projeto , e alterando o projeto final com implementações eliminando as deficiências mantidas na criação doprojeto.

2. Quais são os quatro quadrantes da dívida técnica definidas por Martin Fowler? Defina-as.

   a) Irresponsável e proposital

      O time não tem tempo para o design e utiliza uma solução rápida e com pouca preocupação com a qualidade.

   b) Prudente e proposital

      O time precisa entregar o produto agora com todas as limitações conhecidas e assume de maneira pró-ativa as consequências.

   c) Irresponsável e sem querer

      O time não tem consciência dos princípios básico de design e então nem sequer imagina a bagunça que estão adicionando.

   d) Prudente e sem querer

      Isso é verdade para times com excelentes arquitetos. Eles fornecem uma solução que agrega valor ao negócio, mas depois de completar a solução, eles entendem que a abordagem de design poderia ter sido melhor.

3. De que maneira pagamos juros decorrentes da existência de uma dívida técnica?

   O perigo ocorre quando a dívida não é paga. Cada minuto gasto com falhas conhecidas conta como juros sobre essa dívida. Organizações inteiras podem ficar paradas sob o peso de uma implementação não consolidada.

4. Como você definiria code smells?

   É um termo usado quando há algo de errado com o código, pode ser algum processo que esta demorando para terminar, ou algum possível acoplamento que não está bem visível ou se visível não se tem uma alternativa para desacoplamento.

5. Cite ao menos 5 exemplos ou casos que podem ser caracterizados como dívida técnica.

   - Tamanho de uma classe.
   - Elementos visuais não implementados.
   - Velocidade de execução de taregas.
   - Separação de processos por threads.
   - Acoplamento de código.

6. De que maneira podemos pagar a dívida acumulada? Vale mais a pena pagar ou rolar a dívida?

   Inicialmente executa-se uma analise estática do código fonte, para que suas "dívidas"com o desenvolvimento possam ser mapeadas, em seguida podemos elencar possíveis soluções e estimativas para implementações dessas melhorias ou consertos, assim podemos analisar a real necessidade dessas "dividas"e o tempo disponível para aplica-las, no caso de não haver tempo para implementar todas as soluções, analisar a criação de novas dividas para que o processo de seguimento em novos prazos. Vale salientar que dependendo do software e sua solicitação algumas elencadas "dívidas"podem ser depreciadas com o tempo corrente de sua não implementação.

7. Cite ao menos 3 boas práticas que podemos tomar para evitar a tomada de empréstimo, ou seja, gerar dívidas técnicas.

   a) Aumento da quantidade de testes unitários em regras de negócios.
   b) Aumento da quantidade de testes unitários para cada correção executada.
   c) Correção de bugs de código fonte (ver conceito do SonarQube em https://docs.sonarqube.org/displ
   d) Correção das vulnerabilidades (ver conceito do SonarQube em https://docs.sonarqube.org/display
   e) Correção de code smells (ver conceito do SonarQube em https://docs.sonarqube.org/display/SONA
   f) Redução de código duplicado.
   g) Redução de comentários desnecessários.
   h) Redução da complexidade ciclomática.