Template 1

Este template se destina à primeira parte da atividade, onde os alunos farão a revisão informal e anotarão os defeitos conforme a prática que já tiveram.

Nome do Aluno: Guilherme Gonçalves Marafon, Guilherme Reis Carvalho, Nicolas Amaral Lobo, Vinicius Meier Trevisan

Diagrama UML Analisado: [Diagrama de Casos de Uso / Diagrama de Classes / Diagrama de Sequência]

1. Defeito Encontrado

- **Onde está o defeito:**
  - Diagrama de Casos de Uso.
- **Qual é o defeito:**
  - Caso de Uso ausente para exclusão de avaliação. O requisito funcional RF-16 especifica que "O sistema deve permitir que os usuários deletem inspeções indesejadas". No entanto, não há um caso de uso correspondente no diagrama para representar esta funcionalidade.
- **Sugestão de correção:**
  - Adicionar um novo caso de uso, como "UC13 - Excluir Avaliação", e associá-lo aos atores apropriados (provavelmente Avaliador e Gerente de Produto). Este caso de uso poderia estender o UC08 - Consultar Avaliações.

2. Defeito Encontrado

- **Onde está o defeito:**
  - Diagrama de Casos de Uso.
- **Qual é o defeito:**
  - Fluxo entre solicitação e aceite de avaliação não representado visualmente. O ator Gerente de Produto inicia o UC11 - Solicitar Avaliação, e o Avaliador realiza o UC07 - Aceitar/Recusar Avaliação. Embora a lógica esteja descrita nos requisitos (RF-08, RF-09), o diagrama não mostra nenhuma relação ou dependência entre esses dois casos de uso, o que torna o fluxo do processo menos intuitivo.

- **Sugestão de correção:**
    - Nota de restrição ou uma seta tracejada de dependência poderia ser adicionada entre UC11 e UC07 para indicar que a solicitação é um pré-requisito para o aceite/recusa.

3. Defeito Encontrado

- **Onde está o defeito:**
    - Classe User e UserType.
- **Qual é o defeito:**
    - Modelo de usuário incompleto. O documento descreve três personas importantes: Gerente de Produto, Avaliador Especialista e Avaliador Iniciante. A distinção entre avaliador especialista e iniciante é crucial para o objetivo da ferramenta, que visa auxiliar novatos. O diagrama de classes modela o tipo de usuário (gerente ou avaliador) através da classe UserType, mas não captura a diferença de nível de experiência do avaliador.
- **Sugestão de correção:**
    - Adicionar um atributo na classe Avaliador para representar o nível de experiência.

4. Defeito Encontrado

- **Onde está o defeito:**
    - Classe Problem.
- **Qual é o defeito:**
    - Atributo de "Recomendação" ausente na classe Problema. O requisito RF-06 e a especificação do caso de uso UC03 afirmam que um problema registrado deve conter "sugestões de melhoria" ou "recomendação". No entanto, a classe Problem, conforme inferido do diagrama, parece não possuir um atributo para armazenar essa informação.
- **Sugestão de correção:**
    - Adicionar String à classe Problem.

5. Defeito Encontrado

- **Onde está o defeito:**
    - Associação entre as classes User e Evaluation.
- **Qual é o defeito:**
    - Relacionamento entre User e Evaluation simplificado demais. O texto afirma que um gerente de produto requisita avaliações e um avaliador as executa. O diagrama mostra uma única associação entre User e Evaluation, o que não distingue claramente entre o papel de "solicitante" (Gerente de Produto) e "executor" (Avaliador).
- **Sugestão de correção:**
    - Modelar duas associações distintas partindo da classe Evaluation: uma para o User com o papel de solicitante e outra para o User com o papel de avaliador.

6. Defeito Encontrado

- **Onde está o defeito:**
    - Diagrama de Sequência: Realizar Avaliação.
- **Qual é o defeito:**
    - Título do diagrama inconsistente com o fluxo modelado. O diagrama é intitulado "Realizar Avaliação", mas o fluxo detalhado representa principalmente as operações de CRUD (criar, ler, atualizar, deletar) para um "Problema". O processo de "Realizar Avaliação" é mais amplo, envolvendo a seleção de objetivos e a análise de múltiplas heurísticas, conforme descrito em UC03.
- **Sugestão de correção:**
    - Renomear o diagrama para "Gerenciar Problema" para refletir com mais precisão seu conteúdo, ou criar um diagrama de mais alto nível para "Realizar Avaliação" que invoque as interações de gerenciamento de problemas.

7. Defeito Encontrado

- **Onde está o defeito:**
    - Diagrama de Estado da Avaliação.
- **Qual é o defeito:**
    - Ausência de um estado inicial para avaliações solicitadas. O caso de uso UC11 especifica que uma nova solicitação de avaliação é criada com o status "Pendente" ou "Aguardando Avaliação". O diagrama de estado, no entanto, não possui um estado correspondente. Ele inicia e transita diretamente para "ACEITA" ou "CANCELADA/RECUSADA".
- **Sugestão de correção:**
    - Adicionar um estado inicial chamado "Solicitada" ou "Pendente" após o nó de início, do qual partiriam as transições para "ACEITA" e "CANCELADA/RECUSADA".

8. Defeito Encontrado

- **Onde está o defeito:**
    - Transições do Diagrama de Estado.
- **Qual é o defeito:**
    - Nomenclatura das transições. As transições estão nomeadas com o estado passado (ex: "Avaliação Aceita", "Avaliação Iniciada"). Pelas convenções da UML, as transições devem ser rotuladas com o evento/ação que causa a mudança de estado.
- **Sugestão de correção:**
    - Renomear as transições para verbos no infinitivo que representem a ação, como “AceitarAvaliacao”, “IniciarAvaliacao”, “CancelarAvaliacao” e “Finalizar Avaliacao”.

9. Defeito Encontrado

- **Onde está o defeito:**
    - Estado "CANCELADA/RECUSADA".
- **Qual é o defeito:**
    - "CANCELADA/RECUSADA" é ambíguo. O estado agrupa dois conceitos que ocorrem em momentos distintos do ciclo de vida. Uma avaliação é "Recusada" antes de ser iniciada, enquanto é "Cancelada" após já estar "Em Andamento". Agrupar os dois pode

levar a uma perda de informação sobre o motivo pelo qual a avaliação não foi concluída.

- **Sugestão de correção:**
    - Dividir em dois estados finais distintos: "Recusada" e "Cancelada".