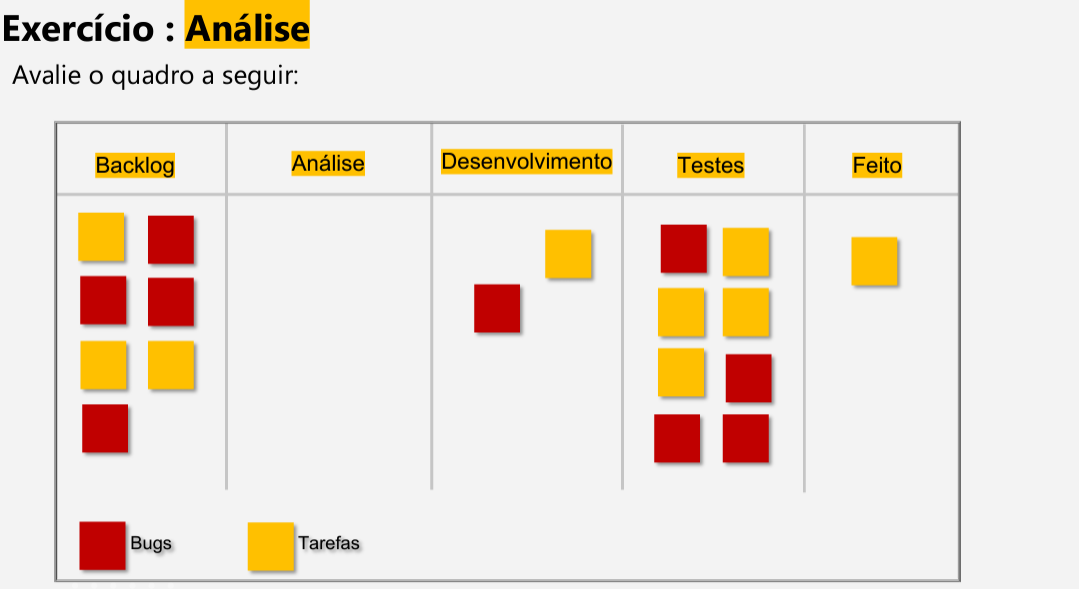
|  |
| --- |
| **– EBAC** |
|  |



Exercício 1: Análise do Kanban

1. **Backlog**:
   * Problema: Excesso de tarefas e bugs no backlog pode dificultar a priorização e o planejamento adequado das atividades.

* Possíveis soluções:
  + Limpar o backlog, priorizando e refinando as tarefas e os bugs com base no valor para o cliente e na criticidade.
  + Implementar um sistema de triagem para bugs, priorizando aqueles que têm maior impacto no usuário ou no sistema.

1. **Análise**:
   * Problema: Falta de clareza sobre o escopo das tarefas e dos bugs em análise, o que pode resultar em retrabalho durante o desenvolvimento.

* Possíveis soluções:
  + Estabelecer critérios claros de aceitação para tarefas e bugs em análise, garantindo que todos na equipe tenham uma compreensão compartilhada do que precisa ser feito.
  + Realizar sessões de revisão de requisitos para eliminar ambiguidades antes que as tarefas entrem em desenvolvimento.

1. **Desenvolvimento**:
   * Problema: Presença de tarefas e bugs em desenvolvimento não balanceada, o que pode causar gargalos e atrasos no progresso geral.

* Possíveis soluções:
  + Balancear o trabalho em progresso limitando o número de tarefas e bugs em desenvolvimento simultâneo.
  + Promover a colaboração entre os membros da equipe para resolver impedimentos e garantir um fluxo contínuo de trabalho.

1. **Testes**:
   * Problema: Acúmulo significativo de bugs aguardando testes, o que indica um possível estrangulamento na equipe de testes ou na cobertura de testes automatizados.

* Possíveis soluções:
  + Aumentar a capacidade de teste expandindo a equipe de teste ou investindo em automação de testes para acelerar o processo de verificação.
  + Introduzir critérios de entrada mais rigorosos para evitar que bugs cheguem ao estágio de testes sem uma validação adequada.

1. **Feito**:
   * Problema: Presença de bugs no estado "Feito", o que sugere que a definição de pronto não está sendo adequadamente aplicada ou que o critério de aceitação está sendo negligenciado.

* Possíveis soluções:
  + Reforçar o critério de aceitação e revisar a definição de pronto para garantir que os itens marcados como "Feito" atendam às expectativas de qualidade.
  + Implementar um processo de revisão final antes de marcar uma tarefa ou bug como concluído, para garantir que todos os requisitos tenham sido atendidos.

**Exercício 2: Papel do QA**

O papel do QA em cada evento ágil é crucial para garantir a qualidade do produto em todas as etapas do processo de desenvolvimento.

1. **Planejamento (Planning)**:
   * O QA pode revisar os requisitos do produto e identificar potenciais problemas de qualidade desde o início.
   * Pode colaborar com os desenvolvedores para garantir que as histórias de usuário sejam claras e testáveis.
   * Estimar esforços de teste para histórias de usuário durante o planejamento.
2. **Daily** **Standup**:
   * Compartilhar atualizações sobre o estado dos testes em andamento.
   * Identificar e relatar qualquer impedimento ou obstáculo relacionado à qualidade.
   * Colaborar com a equipe para priorizar as atividades de teste conforme necessário.
3. **Revisão (Review)**:
   * Demonstrar os casos de teste executados e os resultados obtidos.
   * Fornecer feedback sobre a qualidade das funcionalidades desenvolvidas.
   * Identificar áreas de melhoria nos processos de teste.
4. **Retrospectiva**:
   * Revisar o desempenho dos testes durante o sprint e discutir maneiras de melhorar.
   * Identificar quaisquer padrões de problemas de qualidade recorrentes e propor soluções.
   * Colaborar com a equipe para implementar melhorias contínuas nos processos de teste.
5. **Refinamento (****Refinement)**:
   * Participar na discussão de requisitos e critérios de aceitação das histórias de usuário.
   * Identificar e esclarecer qualquer ambiguidade nos requisitos que possa afetar a qualidade.
   * Propor casos de teste adicionais para garantir uma cobertura completa dos requisitos.

Em resumo, o QA desempenha um papel fundamental em todas as etapas do ciclo de vida do desenvolvimento ágil, ajudando a garantir que a qualidade seja integrada desde o início e mantida ao longo do tempo. Ao colaborar de perto com a equipe de desenvolvimento, o QA contribui para a entrega de um produto de alta qualidade que atenda às expectativas dos usuários.