

**CURSO DE QUALIDADE DE SOFTWARE**

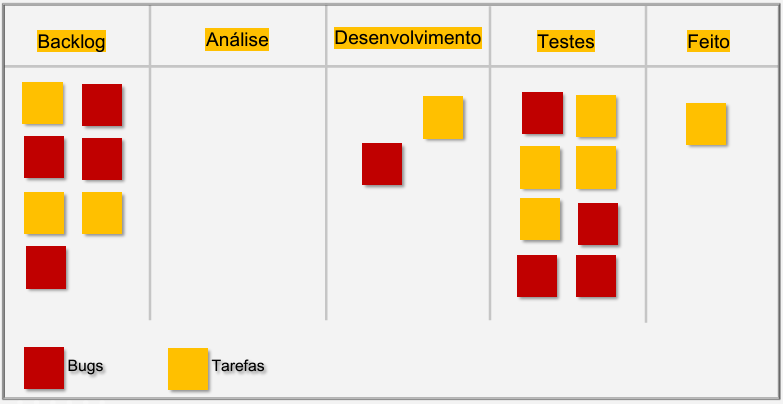
**Guilherme Menezes da Cunha**

Análise de board Kanban

Campinas

2024

**Análise de caso do seguinte board de Kanban**



1. **Quais os problemas você enxerga neste quadro? Liste pelo menos um problema por coluna (backlog, análise, desenvolvimento, testes, feito)**

**R.:** Podemos ver que o quadro expressa um problema grave no **Backlog**, onde temos não só um acúmulo de atividades, mas também temos mais problemas do que tarefas em execução. Podemos extrair dessa análise de backlog o perfil do nosso time, muito provável que apresentem um problema de comunicação e uma dificuldade em entender o objetivo da sprint, ou uma dificuldade em identificar problemas com antecedência e somado a um possível atraso no cronograma de trabalho. O que nos leva ao segundo passo, a **Análise** do produto, onde temos uma coluna esvaziada de tarefas em execução, e como vimos anteriormente no curso, a ausência de bugs não implica na sua não existência, passar apressadamente por esse processo pode nos causar danos muito maiores nas etapas futuras. O que nos passa à terceira coluna, onde temos o **Desenvolvimento** do nosso produto, vemos também um esvaziamento nessa área, o que demonstra mais uma vez uma possível pressa na entrega do produto, onde é realizada apenas uma tarefa, e essa mesma tarefa possivelmente impedida por um bug, nos deixando com a esteira de desenvolvimento travada. Na coluna de **Testes** é onde vemos tarefas e problemas acumulados e inflados devido à não termos ninguém no time responsável pela análise do nosso projeto, nos obrigando a alocar uma grande fatia da equipe para a etapa de testes causada pelo acúmulo de tarefas e de problemas nessa etapa tão importante, a que precede a entrega de nosso produto. Por fim, temos a coluna de tarefas **Feitas** esvaziada, reflexo da esteira de desenvolvimento estar travada por um bug e a etapa de testes com sobrecarga de trabalho e de problemas, o time terá dificuldade na entrega do produto, com apenas uma tarefa entregue, é improvável que esse time esteja dentro do cronograma de entrega deste produto.

1. **Quais as possíveis soluções? Escreva pelo menos 5 soluções.**

**R.:** De imediato precisamos fazer um refinamento do nosso Product **Backlog**, reunir o time e juntos, através de uma reunião, alocar nossos pontos de produtividade e repensarmos alguns story points e traçar uma nova estratégia de MVP (Mínimo Produto Viável), para termos uma ideia de como distribuir nossa equipe para darmos prioridades nos pontos mais críticos do projeto, sem afetar a esteira de produção. O próximo passo é dar mais prioridade à nossa etapa de **Análise** do projeto. Com um backlog já preparado, podemos voltar a ter uma fatia do time direcionado para analisar as tarefas mais prioritárias, será interessante criarmos um board de review das nossas atividades, para podermos extrair ao máximo tarefas que foram realizadas com sucesso, cases que tenham pontos a melhorar e metodologias que não funcionaram ao longo da última sprint. Podemos assim, seguir com a retomada do **Desenvolvimento** do nosso produto, nessa etapa é fundamental realizarmos a tabulação do nosso burndown e burndown épico, assim teremos material com o insight produtivo não só da nossa última sprint, mas também no escopo total do nosso projeto, com todas as métricas de velocidade, controle, fluxo produtivo, tarefas criadas e resolvidas e nosso acumulativo total de desenvolvimento e entrega. Com nossos boards de refinamento de backlog e de burndown feitos, podemos resolver nossa área de **Testes**, o mais prático para o caso que temos de acúmulo de tarefas e problemas seria a aplicação de métricas de DoR (Definition of Ready) e DoD (Definition of Done). No DoR, teremos um refinamento das tarefas que realmente estão prontas para serem realizadas, prepararmos o protótipo do que teremos do nosso projeto, eliminando o acúmulo de tarefas através de uma metodologia INVEST (independente, negociável, valiosa, estimável, small (pequena) e testável). Já no DoD, teremos o escopo qualitativo do que já foi **Feito** e entregue, uma vez que precisamos atender à alguns critérios para considerarmos uma task já finalizada, como cobertura mínima de testes, realização de testes automatizados para checagem dos critérios de aceitação do usuário, checagem da conformidade de toda a documentação das funcionalidades implementadas e health check da eficiência e eficácia da produção. Com um review de projeto mais aprofundado com o time, aplicação das metodologias ágeis, implementação de MVP e Burndown, DoR e DoD, conseguimos reestruturar nossa forma produtiva, reconhecer histórias que não funcionam e gerenciar o projeto de forma a atender todos os critérios de qualidade e conformidade e cumprindo com o cronograma esperado para o projeto.

**2.2 Sugestões de tarefas para o QA em cada evento ágil:**

**R.:** Considerando que 6 eventos ágeis principais tenham sido utilizados por esse time: Sprint Planning, Daily Meeting, Sprint Review, Retrospective, Sprint Refinement e Behavior Driven Development, é de suma importância que o QA esteja presente em todas as etapas, se comunicando de forma adaptativa com o time de acordo com o evento do momento.

**2.2.1 Sprint Planning**

**R.:** Aqui o profissional de QA precisa trabalhar com o time para que cheguem à uma definição clara e factível de que os critérios de aceitação estão de fato dentro dos parâmetros colocados pelos story points do usuário, uma vez que essas histórias também tenham sido avaliadas pelo QA, que apresentará uma análise de risco de cada story point, trará uma estimativa de tempo mais realista da execução das tarefas e dos percalços que possam aparecer durante os testes.

**2.2.2 Daily**

**R.:** A principal função do QA na Daily é passar ao time qual o status dos testes atuais, elencando quaisquer bloqueios que possam impactar na entregas dos story points propostos, permitindo que o time responda de forma mais rápida e mantenha a sprint no cronograma estipulado, podendo também o profissional de QA sugerir melhorias contínuas baseadas no material discutido, incluindo ajustes nas práticas de testes para aumento da eficácia e eficiência no desenvolvimento do projeto.

**2.2.3 Sprint Review**

**R.:** Aqui é hora do profissional de QA dar o parecer qualitativo do material desenvolvido até o momento baseando-se nas sprints anteriores, apontando pontos que deram certo e áreas que precisam de melhorias, para que assim o time saiba planejar as próximas sprints a atender futuras melhorias, seria interessante nessa etapa o QA apresentar uma demonstração dos testes realizados até o momento, mostrando automação de teste de funções recém adicionadas e mostrando como esses testes garantem uma qualidade elevada e identificam problemas antes da entrega ao usuário, aumentando a confiança no projeto entre os stakeholders.

**2.2.4 Retrospective**

**R.:** Na retrospectiva o QA deverá contribuir com a identificação de melhorias nos processos de garantia de qualidade do projeto apresentando métricas de qualidade, como taxa de defeitos encontrados e resolvidos, cobertura dos testes e velocidade de resolução de bugs encontrados. Essas métricas apoiam o QA em sugerir novas ferramentas, técnicas ou práticas que possam aumentar a eficiência e eficácia dos testes nas próximas sprints baseando-se nas tendências de melhoria contínua do processo de desenvolvimento e QA.

**2.2.5 Sprint Refinement**

**R.:** No refinamento o QA pode ajudar a equipe a particionar os story points do usuário em etapas menores e mais gerenciáveis, garantindo que os critérios de aceitação estejam bem definidos, facilitando o trabalho de desenvolvimento e testabilidade, onde todos os aspectos para cobertura total dos testes sejam abrangidos, via testes de limite de valor e casos de testes negativos.

**2.2.6 BDD**

**R.:** Como agente ativo entre stakeholders e Dev Team, o QA terá estreita colaboração entre as partes citadas para escrever os cenários de testes baseado nos story points delimitados usando a linguagem Gherkin, garantindo um entendimento comum dos critérios de aceitação e que os testes automatizados tenham cobertura total esperada, podendo serem validados conforme atendam não só as expectativas do projeto, mas que também sejam executados de forma mais eficaz.