

# Planejamento de Testes – SQUAD 2

## 1. Escopo

O escopo é a etapa em que o time define claramente o que será testado e o que ficará fora dos testes. Tudo deve ser documentado para que o cliente saiba exatamente o que será avaliado. Em resumo, devemos:

Realizar uma reunião para decidir o que precisa ou não ser testado, priorizando as funcionalidades mais críticas do software.

Registrar de forma objetiva o que fará parte do teste e o que ficará excluído.

---

## 2. Cronograma

Precisamos estabelecer datas para a entrega dos resultados de cada teste e compreender o cenário do cliente, avaliando se eventuais atrasos são aceitáveis. Assim, conseguimos definir o prazo final de entrega do software. Em resumo, devemos:

- Definir o prazo para a execução de cada teste.
  - Verificar se o prazo de entrega do software é negociável ou não.
  - Determinar a data prevista para a entrega do software.
- 

## 3. Orçamento

A elaboração de um orçamento de QA inicia pela definição do que será testado, do que ficará fora do escopo e de quais objetivos de qualidade o time pretende atingir. Essa etapa ajuda a priorizar riscos e a direcionar o esforço para as áreas mais críticas do sistema. Em seguida, é necessário identificar os recursos indispensáveis para a operação de testes, como o tamanho da equipe, o tipo de profissionais, as ferramentas que precisam ser utilizadas e a infraestrutura necessária para simular corretamente o ambiente de produção.

Com o escopo e os recursos bem definidos, o próximo passo é estimar o esforço e o tempo necessários para executar as atividades de teste, considerando os diferentes níveis (unitário, integração, sistema, aceitação) e tipos de testes (funcionais e não funcionais). Pode-se utilizar técnicas para dividir as tarefas em partes menores, onde o esforço é calculado de maneira mais objetiva com base no tamanho e na complexidade de cada item. Após isso, calcula-se o custo total combinando gastos diretos, como salários, ferramentas e equipamentos, e custos indiretos ligados à operação e manutenção do ambiente de QA.

### 1. Estrutura do orçamento

- Escopo do que será testado e objetivos de qualidade.
- Riscos críticos do sistema e priorização.
- Recursos humanos, ferramentas e infraestrutura necessários.

### 2. Estimativa de esforço

- Horas ou story points previstos para cada etapa.
- Consideração dos tipos e níveis de teste.
- Uso de dados históricos quando disponíveis.
- Aplicação da técnica para calcular o esforço de forma mais objetiva.

### 3. Cálculo financeiro

- Custos diretos: equipe, licenças, equipamentos.
  - Custos indiretos: operação, manutenção, infraestrutura compartilhada.
- 

## 4. Qualidade

Para avaliar a qualidade no planejamento de testes, é fundamental definir indicadores claros que retratem o estado atual do software e sua evolução durante o ciclo. Esses indicadores geralmente incluem o volume de defeitos identificados, sua distribuição por área do sistema, a severidade média e o comportamento das falhas ao longo do tempo. Medir o intervalo entre encontrar e resolver um bug também é relevante, pois reflete a eficiência do time e a maturidade do processo. A análise consistente desses dados permite identificar quais partes do produto são mais instáveis e onde concentrar esforços de prevenção e testes.

A partir desses indicadores, estabelecem-se metas de qualidade que orientam o trabalho do time. Tais metas funcionam como parâmetros mensuráveis de melhoria como, por exemplo, reduzir a média de bugs por história, diminuir o tempo de correção, aumentar o percentual de testes aprovados ou melhorar a estabilidade das entregas antes de chegarem ao QA. São objetivos práticos, que direcionam ações mais eficazes no ciclo. Quando bem definidas, tornam o planejamento mais realista e fortalecem o alinhamento entre desenvolvimento e QA.

### Indicadores de Qualidade

- Quantidade total de bugs por ciclo e por área do software.
- Severidade e criticidade média das falhas.
- Tendência de aumento ou redução de defeitos ao longo das sprints.
- Tempo médio entre detecção e correção de bugs (MTTR).

- Percentual de retrabalho causado por problemas de qualidade.

## **Metas de Qualidade**

- Reduzir a média de bugs por história.
  - Diminuir o tempo de correção de falhas.
  - Aumentar a taxa de aprovação dos testes.
  - Melhorar a qualidade das entregas antes de chegarem ao QA.
- 

## **5. Riscos**

A etapa de riscos identifica tudo que pode comprometer a qualidade do software ou atrapalhar o cronograma. O objetivo é antecipar problemas e definir como tratá-los caso aconteçam. Entre os principais riscos estão atrasos no desenvolvimento, mudanças de requisitos, instabilidade dos ambientes de teste, dependências externas não entregues e falta de recursos técnicos ou humanos. Também é importante considerar riscos ligados à comunicação entre times e ao nível de conhecimento da equipe.

### **Principais riscos**

- Atraso nas entregas necessárias para iniciar o QA.
- Mudanças de requisitos que afetam escopo e prazos.
- Ambiente de homologação instável ou indisponível.
- Dependências externas atrasadas.
- Falta de ferramentas, dispositivos ou recursos adequados.
- Falhas de comunicação entre áreas.

### **Mitigação**

- Alinhamentos frequentes com as equipes.
  - Controle formal de mudanças.
  - Solicitação antecipada de ambientes e ferramentas.
  - Planos de contingência para atrasos.
- 

## **6. Pessoas**

A etapa de pessoas define quem participa do processo, o que cada um faz e como as responsabilidades são distribuídas. Isso evita confusão, melhora a comunicação e garante que o ciclo de testes aconteça de forma organizada.

## 6.1 Funcionários

Os funcionários internos são responsáveis pela execução principal dos testes. Entre eles estão:

- Analistas de QA que criam e executam casos de teste.
- Automação, quando necessário.
- Coordenador ou líder que organiza tarefas e faz o elo com outros times.
- Desenvolvedores que corrigem bugs e apoiam o QA.

## 6.2 Terceiros

Terceiros são utilizados quando a equipe interna não possui tempo ou especialização suficiente.

Incluem consultorias para testes específicos, reforço temporário ou serviços externos como farms de dispositivos. É importante definir escopo, responsabilidades e prazos.

## 6.3 Necessidades de Treinamento

Treinamentos servem para nivelar o time e garantir que todos saibam usar as ferramentas e práticas do projeto. Podem incluir:

- Uso de novas ferramentas.
- Metodologias ágeis.
- Testes específicos (performance, segurança).
- Padrões internos da empresa.

---

## 7. Aquisições

O planejamento de aquisições no contexto de testes envolve identificar e gerenciar a compra ou contratação de tudo o que é externo à empresa, mas vital para a execução da garantia de qualidade. Isso não se limita apenas a comprar computadores envolve licenças de softwares específicos (ferramentas de automação, gestão de testes, virtualização), dispositivos móveis reais para testes de compatibilidade, contratação de consultorias especializadas ou serviços de nuvem para ambientes de homologação. É crucial que o time de QA mapeie essas necessidades com antecedência, pois processos de compra corporativa costumam ser burocráticos e demorados. Se uma ferramenta essencial não estiver disponível no início da Sprint, todo o cronograma de testes pode ser comprometido. Além disso, deve-se avaliar o custo-benefício entre construir uma solução interna (ferramentas open source) versus adquirir uma solução de mercado (ferramentas enterprise).

Itens de Aquisição e Infraestrutura

- **Licenciamento de Software:** Ferramentas de gestão (Jira, TestRail), automação (SaaS) ou IDEs pagas.

- **Hardware e Dispositivos:** Celulares, tablets ou servidores específicos para simular o ambiente do usuário.
  - **Serviços em Nuvem:** Ambientes temporários na AWS/Azure ou farms de dispositivos (ex: BrowserStack, SauceLabs).
  - **Contratação de Terceiros:** Consultoria especializada ou outsourcing para testes de carga/segurança, se a equipe interna não possuir a expertise.
- 

## 8. Comunicação

O plano de comunicação é a ferramenta que garante a transparência do processo de QA, definindo como, quando e para quem as informações sobre a qualidade do software serão transmitidas. Um dos maiores riscos em projetos de software é a falha na comunicação, onde stakeholders não sabem o status real dos bugs ou desenvolvedores não recebem feedback rápido. Nesta etapa, define-se a Matriz de Comunicação, que estabelece a rotina de feedbacks: reuniões diárias (Dailies), relatórios de bugs, Dashboards de métricas em tempo real e o relatório final de encerramento de testes. O objetivo é evitar o "efeito surpresa" no final do projeto e garantir que todos (Product Owner, Devs e Gerentes) estejam alinhados sobre os riscos e a estabilidade da entrega.

### Estrutura da Comunicação

- **Público-alvo (Quem):** Definir quem precisa receber cada tipo de informação.
- **Canais (Onde):** Ferramentas oficiais (E-mail, Teams, Confluence e Jira).
- **Periodicidade (Quando):** Frequência dos reportes (Diário, Semanal, Por Sprint, Na ocorrência de erro crítico).
- **Artefatos (O Quê):**
  - *Status Report:* Resumo do progresso da execução dos testes.
  - *Bug Report:* Documentação detalhada das falhas encontradas.
  - *Relatório de Encerramento:* Parecer final sobre a qualidade da versão para Go-Live.