

Teste de habilidades

Dono de Produto

Descrição

Imagine que estamos iniciando um novo trimestre na empresa e que você é o Dono de Produto da *squad*. O Gerente de Produto responsável explicou que os usuários estão com dificuldade em finalizar o processo de *onboarding* e que têm reclamado da sua fragmentação. Com isso em mente, ele apresentou os principais indicadores para o período e, agora, precisa que você faça o melhor que puder para entregar os resultados esperados.

Indicadores

- Reduzir em 5% a taxa de saída¹ do processo de *onboarding* em relação ao período anterior;
- Aumentar em 10% a taxa de ativação em relação ao período anterior;

Premissas

1. Utilize o aplicativo da empresa como balizador das suas decisões;
2. Cada sprint tem *timebox* de 2 semanas;
3. A equipe utiliza Scrum como *framework* de desenvolvimento;
4. Cada estória corresponde a um cartão no quadro de desenvolvimento (Kanban);
5. Sua equipe de desenvolvimento é composta por 5 desenvolvedores.

¹ Considere a taxa de saída como a taxa média de saída de todas as telas percorridas durante o processo de *onboarding*.

Perguntas

Com base neste cenário, responda as perguntas a seguir:

1. **Com as informações disponíveis até o momento, quais melhorias você proporia ao aplicativo?**

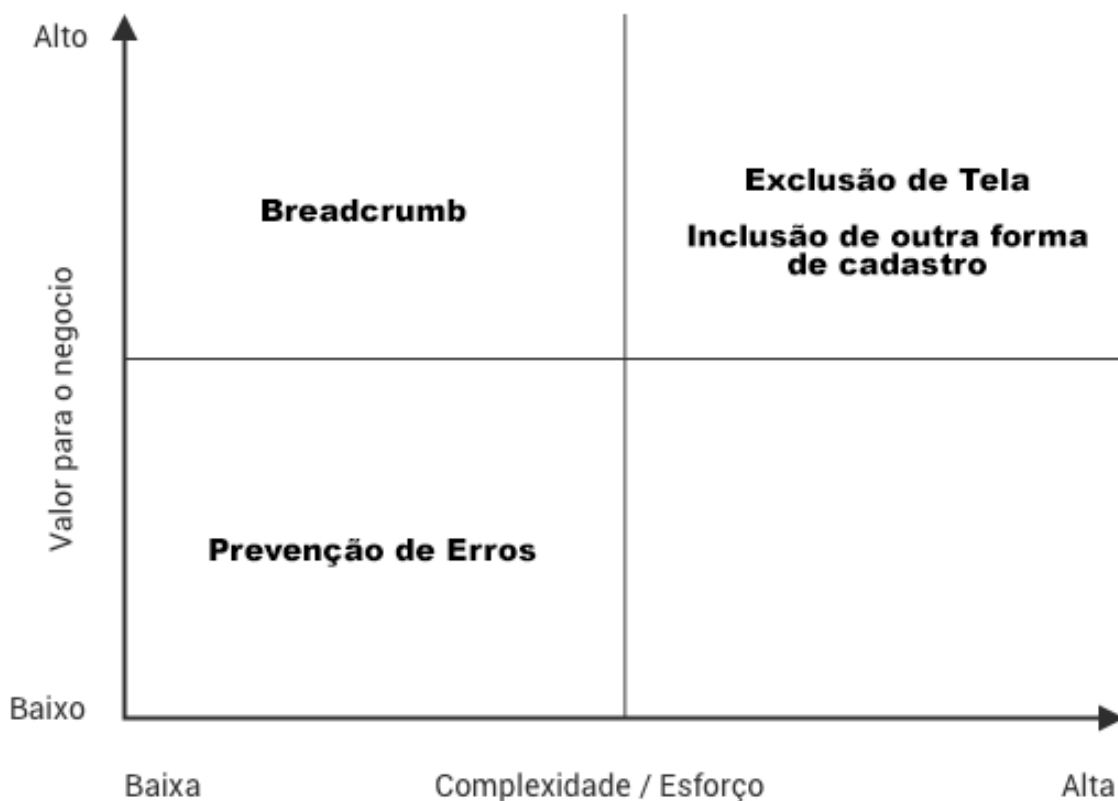
Com o que eu observei do fluxo de onboarding atual, eu sugiro algumas melhorias, baseado nas Heurísticas de Nielsen, que podem melhorar a experiência do usuário e diminuir a taxa de saída nessas telas, como:

- **Inclusão de breadcrumb (fluxo de progresso) em todas as telas, contendo quantos são os steps de onboarding e em qual o usuário está.**
A primeira heurística de Nielsen trata sobre a visibilidade do estado do sistema e é de extrema importância que o usuário saiba em qual ponto do onboarding ele se encontra e quantos são os passos totais para que ele não se sinta perdido neste processo.
- **Exclusão da tela de inserção de e-mail.**
*A oitava heurística trata sobre um design e estética minimalista. Neste momento precisamos do mínimo de informações necessárias para cadastrar o usuário na plataforma e quanto maior o número de informações e telas que solicitamos para o usuário aqui, dificultamos o seu caminho até a entrega de valor (o acesso ao aplicativo). Além disso, o e-mail é novamente solicitado na tela 1 de completar cadastro.
Obs.: Poderíamos reaproveitar essa tela de inserção de e-mail, para realizar o próximo item.*
- **Inclusão de outra forma de cadastro.**
A terceira heurística fala sobre a liberdade e controle do usuário, portanto acho que seria importante, na primeira tela do onboarding, perguntar ao usuário se ele deseja se cadastrar por celular ou por e-mail, mas nunca obtermos ambas as informações neste primeiro momento (pelo motivo explicado no item anterior).
- **Prevenção de erros no cadastro.**
A quinta heurística diz respeito a prevenção de erros, então seria interessante utilizarmos formas de evitar que um erro no cadastro aconteça. No caso de um cadastro via celular, seria interessante realizarmos validações de DDD, número válido, apresentar somente o teclado numérico e no caso de um cadastro via e-mail podemos apresentar uma lista para autopreenchimento do provedor de e-mail, apresentar o teclado alfanumérico, solicitar a confirmação do e-mail

2. Com essas melhorias em mente, como você organizaria o *backlog* de produto? Quais critérios/técnicas utilizaria para priorizá-lo?

Acredito que o primeiro passo seria realizar um refinamento técnico, para entender qual seria o esforço necessário para a realização de cada uma dessas atividades. Neste momento não seria necessário estimar o tempo exato para a construção, mas levantar qual o tamanho das alterações e os impactos dessas mudanças. Com base nestas informações, eu realizaria a priorização dos itens do backlog, avaliando quais gerariam mais valor ao usuário.

Um framework que poderia ser utilizado para a priorização seria a matriz de **Valor x Esforço**. Ela me ajudaria a entender quais itens possuem um grande valor, e o esforço para serem desenvolvidos.



De acordo com a matriz acima, a inclusão de um **breadcrumb** seria uma solução que geraria um grande valor para o usuário e seu esforço e complexidade seriam baixos, talvez sendo possível finalizá-lo em uma sprint.

A exclusão da tela de e-mail e a inclusão de outra forma de cadastro, são itens que alteram diretamente o fluxo de onboarding, portanto acredito que sua complexidade seja maior. Essas duas demandas podem ser entregues de maneira subsequente, em uma sprint tiramos a tela de inclusão de e-mail e em outra sprint atuamos na inclusão de outra forma de cadastro.

E por último, atuaríamos na prevenção de erros, pois é um item que gera um menor valor percebido ao usuário, mesmo possuindo uma complexidade mais baixa.

3. **Selecione um dos itens do seu *backlog* e descreva-o de forma detalhada;**

Item: Inclusão de Breadcrumb

Contexto: Identificamos que os usuários estão com dificuldade em finalizar o processo de onboarding e que têm reclamado da sua fragmentação, por este motivo decidimos incluir um breadcrumb (fluxo de progresso) no topo da tela, contendo quantos são os passos do onboarding e em qual o usuário está.

User Story: **Eu** como usuário **quero** saber quantos passos o processo de onboarding possui e em qual deles eu estou **para que** não me sinta perdido neste processo e identifique as ações com transparência.

Benefícios: Esperamos reduzir em 5% a taxa de saída do processo de onboarding em relação ao período anterior.

Teste: Cenário – O usuário não possui uma conta e clica em criar conta.

Dado que o usuário entrou no fluxo de onboarding **quando** ele estiver navegando por suas paginas **então** ele deverá conseguir identificar em qual passo está e quantos passos são.

Critérios de Aceite:

- Quando o usuário entra no fluxo de onboarding, no topo da tela é apresentado quantos são os passos e em qual ele está.
- Quando o usuário avança no fluxo de onboarding, o breadcrumb deverá atualizar o passo que o usuário avançou.
- Quando o usuário retrocede no fluxo de onboarding, o breadcrumb deverá atualizar o passo para qual o usuário retornou.

4. **Como você planejará as entregas? Quais etapas teriam? Como faria a composição das *sprints*? Quais ferramentas e/ou técnicas utilizaria?**

O planejamento das entregas teria como base o **framework Scrum**.

Antes de iniciar uma **sprint**, eu realizaria um **refinamento técnico** para estimar o esforço de cada item do backlog e após isso faria a priorização desses itens conforme explicado na pergunta 2 deste case.

Com o **backlog priorizado**, realizaria uma **sprint planning** para que a equipe pudesse pontuar as histórias e definirmos o que seria entregue naquela sprint.

Importante ressaltar que os itens de uma mesma funcionalidade que são “blockers” tem prioridade, por exemplo, a equipe de backend precisa finalizar um código para que a equipe de frontend realize a integração e testes de uma determinada funcionalidade, desta maneira, o processo de backend pode iniciar uma sprint antes.

Após definidos os itens, a sprint é iniciada e acompanhada diariamente pelas reuniões de **daily**, e ao término do timebox (2 semanas) realizaríamos uma **sprint review**, para que sejam apresentados os itens concluídos e avaliados os critérios de aceite, além de nos ajudar a determinar futuras adaptações nas funcionalidades entregues. E por fim, realizaríamos uma **retrospectiva da sprint**, para entendermos todos os pontos positivos daquela sprint e tudo o que precisa ser melhorado para as próximas.

5. **Quais sessões você facilitaria e como você faria para que as pessoas do seu time entendam o objetivo que você planejou? Como seria o acompanhamento destas entregas com o time?**

O **refinamento técnico** seria a primeira sessão que eu como Product Owner deveria facilitar, para apresentar as alterações para toda ou parte da equipe (um líder técnico talvez) e juntos entendermos qual o impacto daquelas mudanças.

A **sprint planning** seria a segunda cerimônia que eu facilitaria, para apresentar os itens que precisam ser desenvolvidos para o restante da equipe, já com as **User Stories** escritas, para que a equipe possa pontuar cada item e montar o sprint backlog.

Para realizar o acompanhamento das entregas, eu poderia participar das **daily**s do time como ouvinte e acompanhar o gráfico **burndown** da sprint, para entender se a produtividade está acima ou abaixo do esperado.

Me colocaria também sempre à disposição do time para tirar quaisquer dúvidas relacionadas aos itens e ao produto como um todo.

6. **Quais indicadores você utilizaria para acompanhar a qualidade do processo de desenvolvimento?**

Um dos principais indicadores que eu acompanharia, seria o número de bugs encontrados pelo time de QA durante os testes após o desenvolvimento. Após implantada a solução, nós poderíamos analisar o número de bugs encontrados pelos desenvolvedores em ambiente de homologação e produção.

O gráfico de burndown também nos ajuda a entender a qualidade do processo de desenvolvimento e qual a velocidade/produtividade do time.

7. Quais são os fatores de sucesso para manter o time engajado no objetivo e performance adequados durante todo o desenvolvimento?

Acho que é extremamente importante que todo o time tenha a visão dos motivos pelos quais estamos realizando uma alteração em nosso produto ou funcionalidade.

Poderíamos trabalhar com OKR's e sendo assim, os objetivos chave seriam sempre claros para toda a equipe e os indicadores acompanhados diariamente.

Neste caso, os KPI's que acompanharíamos seriam as taxas de saída e as taxas de ativação, para compreendermos o sucesso das alterações.