



FUNDAMENTOS DE DESIGN DE SISTEMAS

AULA 6



Prof. Winston Sen Lun Fung



CONVERSA INICIAL

Nesta etapa de estudos, vamos discutir sobre conceitos, práticas e estratégias fundamentais que orientam o desenvolvimento de produtos de software centrados no usuário. A importância do design no desenvolvimento de software é inegável, sendo a ponte entre a tecnologia e o usuário, tornando as soluções digitais acessíveis, intuitivas e agradáveis.

Nossos estudos serão guiados pela experiência do usuário (UX), um componente crucial que define como um usuário se sente ao interagir com um sistema. Por meio do UX, nosso objetivo é entender as necessidades, expectativas e comportamentos dos usuários, buscando sempre criar experiências positivas, satisfatórias e significativas.

A funcionalidade e a usabilidade são dois aspectos que andam lado a lado com UX e UI. A funcionalidade diz respeito à capacidade de um sistema executar as tarefas que se propõe a realizar, enquanto a usabilidade avalia a eficácia, a eficiência e a satisfação com que os usuários podem atingir seus objetivos ao usar um sistema.

Acreditamos que o sucesso no desenvolvimento de software é alcançado quando essas disciplinas trabalham em harmonia. Combinando UX, UI, funcionalidade e usabilidade, somos capazes de criar produtos de software que não apenas atendem aos requisitos técnicos, mas também criam experiências profundamente satisfatórias para os usuários.

TEMA 1 – ARQUITETURA DA INFORMAÇÃO

O Information Architecture Institute¹ define a Arquitetura de Informação (AI) como a “prática de decidir como organizar as partes de alguma coisa de modo a torná-la compreensível”.

A AI envolve várias considerações, incluindo a estruturação de uma hierarquia de informação clara e intuitiva, rotulando informações eficazmente e desenvolvendo esquemas de busca eficazes. É essencial para websites ou aplicativos com grandes volumes de conteúdo, em que uma organização de informação mal executada pode tornar difícil para os usuários encontrar o que estão procurando.

¹ Disponível em: <<https://www.iainstitute.org/what-is-ia>>. Acesso em: 2 ago. 2023.



A AI tem como objetivo responder às quatro questões que um usuário pode ter ao entrar em um site ou em uma tela, conforme detalhado por Steve Krug em seu livro *Don't make me think*:

1. O que é isto?
2. O que eles têm aqui?
3. O que posso fazer aqui?
4. Por que devo estar aqui e não em outro site/tela?

A maneira como as informações são agrupadas e disponibilizadas ao usuário é fundamental para fornecer uma navegação de qualidade e uma boa usabilidade. Além disso, a AI ajuda a garantir que as informações sejam consistentes e coerentes em todo o sistema, melhorando a experiência do usuário e reduzindo a frustração.

Portanto, antes de iniciar o desenvolvimento de um produto digital, é crucial considerar a AI. Em outras palavras, como podemos organizar todas as informações de maneira que o usuário possa compreendê-las de forma intuitiva e eficaz?

Em um jogo, podemos pensar nas informações colocadas em um cenário, tudo que ele comunica, as impressões de que ele passa ao jogador, as sensações, percepções e a inserção na história.

Da mesma forma, em um site é preciso projetar como será a jornada do usuário e quais informações serão apresentadas e são importantes para a experiência do usuário.

1.1 Pilares da arquitetura de informação

A AI é uma disciplina essencial no campo do design de experiência do usuário (UX). Ela desempenha um papel fundamental na organização, estruturação e rotulação de informações de maneira intuitiva e fácil de entender para os usuários. Para compreender melhor a AI, precisamos nos familiarizar com seus pilares fundamentais. São eles: navegação, busca, organização e rotulação.

Cada um desses pilares desempenha um papel crítico na criação de uma experiência do usuário eficaz, fornecendo aos usuários uma compreensão clara do que podem fazer em um site ou aplicativo e como podem fazer. Juntos, eles facilitam a exploração, a compreensão e o uso eficaz das plataformas digitais,



permitindo que os usuários encontrem o que estão procurando com facilidade e eficiência.

1. **Navegação:** trata de como os usuários navegam por um site ou sistema. Uma navegação bem projetada permite que os usuários encontrem o que estão procurando com facilidade e eficiência.
2. **Busca:** complementa a navegação, proporcionando aos usuários um meio eficaz de buscar informações específicas dentro do sistema.
3. **Organização:** envolve a organização das informações de maneira lógica e intuitiva, facilitando a localização do conteúdo pelos usuários.
4. **Rotulação:** relaciona-se com a forma como as informações são nomeadas e categorizadas. Rótulos eficazes são intuitivos e refletem a linguagem e a compreensão do usuário.

1.2 Como documentar: documentando a arquitetura de informação

Documentar a AI é um passo crucial que oferece uma compreensão clara do layout e da estrutura de um site ou aplicativo. A documentação pode assumir várias formas, dependendo do projeto e dos objetivos. Aqui estão algumas maneiras comuns de documentar a AI:

1. **Mapas do site:** fornecem uma visão geral da organização do conteúdo em um site ou aplicativo.
2. **Wireframes:** esboços básicos que ilustram a disposição dos elementos em uma página ou tela.
3. **Diagramas de fluxo do usuário:** mostram o caminho que um usuário pode percorrer em um site ou aplicativo, incluindo ações do usuário, pontos de decisão e interações com a interface.
4. **Blueprints ou especificações da página:** detalham o conteúdo e a funcionalidade de uma página específica, incluindo informações sobre interações do usuário e comportamento da interface.
5. **Modelos de conteúdo:** definem os tipos de conteúdo que serão exibidos em um site ou aplicativo e as relações entre eles.
6. **Taxonomias e etiquetagem:** a taxonomia é a classificação das informações e a etiquetagem é como essas informações são representadas para o usuário.



7. Protótipos: são versões interativas do design que permitem testar a AI e a usabilidade antes do desenvolvimento final.

É importante notar que a documentação de AI é um recurso vivo que deve ser atualizado à medida que o sistema evolui. É fundamental para garantir que o sistema seja desenvolvido em conformidade com as necessidades dos usuários e os requisitos do negócio. Ao documentar a AI, é importante considerar o usuário. A documentação deve, portanto, refletir uma estrutura e organização que atendam às necessidades dos usuários, tornando mais fácil para eles localizar e compreender as informações.

No próximo tópico, daremos uma atenção especial aos wireframes e blueprints, que são ferramentas fundamentais na criação de uma AI eficaz.

TEMA 2 – WIREFRAMES

Wireframes são representações visuais simplificadas que surgem durante e após a etapa de Arquitetura de Informação, desempenhando um papel crucial no processo de design da interface do usuário (UI) e na estruturação da informação.

Os wireframes, essencialmente, estabelecem a base para a construção da interface de um site ou aplicativo. Funcionam como uma espécie de “esqueleto” do projeto, definindo a estrutura, o layout e a hierarquia dos elementos de uma página, sem se preocupar, inicialmente, com detalhes gráficos, cores ou estilos. Seu principal objetivo é estabelecer a funcionalidade e o fluxo de interação do usuário no ambiente digital.

Essas representações visuais ajudam a organizar a informação de maneira hierárquica e visual, fornecendo uma visão preliminar do problema a ser abordado. Eles permitem um debate focado sobre quais informações serão apresentadas, seu posicionamento, o conteúdo e algumas funcionalidades, antes mesmo de se entrar em uma fase de exploração mais criativa e detalhada do design.

O wireframe pode ser desde um esboço feito à mão até um desenho mais detalhado feito em uma aplicação especializada. Eles apresentam um layout básico, fornecendo uma visão geral do conteúdo, posicionamento e da estrutura básica de como uma página ou tela será apresentada. No entanto, vale ressaltar



que eles não costumam incluir cores, fontes e elementos de design final, já que o objetivo principal é focar na funcionalidade, estrutura e fluxo de informação.

Os wireframes servem como uma ferramenta eficaz para validar ideias e conceitos iniciais, promovendo uma comunicação clara e eficaz entre designers, desenvolvedores e stakeholders, e permitindo ajustes e iterações antes que o processo de design detalhado e codificação comece. Eles representam um passo fundamental no processo de desenvolvimento, economizando tempo, recursos e esforço, ao minimizar mal-entendidos e reestruturações drásticas mais adiante no processo de design.

2.1 Construindo o wireframe

Wireframes são, essencialmente, uma representação visual esquemática de uma interface. Seu objetivo é trazer a estrutura, a hierarquia de informações e a funcionalidade do layout de uma página, sem a necessidade de se preocupar com cores, tipografias e elementos gráficos.

Por exemplo, suponha que você está projetando a página de um e-commerce. No seu wireframe, você pode começar esboçando os principais blocos de conteúdo: um cabeçalho com o logo e a barra de pesquisa, uma navegação principal, um banner promocional, uma lista de produtos, um rodapé com informações adicionais etc. Em seguida, dentro desses blocos, você definirá quais elementos eles incluirão: títulos, textos, botões, links, imagens e assim por diante.

Ao entregar um wireframe para a equipe ou para o cliente, a intenção é que todos possam ter uma visão clara e compartilhada de onde o projeto está caminhando. Esse artefato ajuda a garantir que todos estejam na mesma página e tenham uma compreensão comum do que está sendo planejado, minimizando possíveis desentendimentos ou desvios futuros.

Por exemplo, em um projeto de redesign de um site de notícias, um wireframe pode ajudar a equipe a visualizar a nova disposição dos artigos na página, a nova estrutura de categorias, os locais para publicidade etc., antes mesmo de qualquer código ser escrito ou de qualquer elemento visual ser criado.

Um dos aspectos mais importantes a se ter em mente ao criar wireframes é a usabilidade. É essencial que o wireframe não apenas apresente um layout organizado e esteticamente agradável, mas que também promova uma experiência do usuário eficiente e intuitiva.



Por exemplo, ao projetar o wireframe para um formulário de inscrição, você deve considerar a ordem e o agrupamento das perguntas, o tamanho e a localização dos campos de entrada, os elementos de ajuda e feedback, entre outros fatores que influenciam na facilidade de uso do formulário.

Por último, lembre-se de que os wireframes são ferramentas de comunicação. Eles são usados não apenas para esboçar suas ideias, mas também para explicá-las aos outros. Portanto, certifique-se de que seu wireframe seja compreensível para todos que o verão, incluindo clientes, outros designers, desenvolvedores e qualquer outra pessoa envolvida no projeto.

Por exemplo, ao criar um wireframe para um aplicativo de reservas de restaurantes, você pode usar anotações e descrições detalhadas para indicar como as interações funcionarão, como a busca, a seleção de restaurantes, a visualização de detalhes, o processo de reserva e assim por diante.

2.2 O que incluir em um wireframe?

Ao criar um wireframe organizado, simples e limpo, é essencial concentrar-se em vários elementos fundamentais. São eles:

- **Conteúdo:** o conteúdo refere-se à informação que se deseja exibir na tela. O objetivo é responder à pergunta: “o que será exibido nesta tela?”. O wireframe ilustra onde cada pedaço de conteúdo será posicionado na página, podendo incluir áreas para texto, imagens, botões e muito mais. Por exemplo, em um site de notícias, o conteúdo pode incluir manchetes, imagens, textos de notícias, links para notícias completas etc.
- **Estrutura:** a estrutura diz respeito à maneira como os componentes menores vão compor o wireframe. O objetivo é responder à pergunta: “como os elementos desta página serão reunidos?”. Um wireframe normalmente mostra onde os elementos de navegação (como menus, barras de pesquisa e links) serão colocados, ajudando a definir o fluxo de como um usuário vai interagir com o site ou aplicativo.
- **Hierarquia:** a hierarquia anda de mãos dadas com a estrutura. Aqui, definimos quais elementos da estrutura merecem maior destaque, sempre focando na informação que desejamos transmitir. Por exemplo, em um e-commerce, um título de produto pode ter mais destaque do que a



descrição do produto. O objetivo é responder à pergunta: “o que você quer que o usuário veja primeiro?”. Devemos levar em consideração:

- O que tem mais valor para o usuário?
- O que tem menos valor para o usuário?
- Como esses elementos serão exibidos, agrupados e associados?
- **Funcionalidade:** ao construir wireframes, é crucial considerar as funcionalidades básicas que estarão associadas a ele. A pergunta a ser respondida é: “como esses elementos funcionarão juntos?”. Os wireframes podem também incluir descrições de como certos aspectos da página funcionarão. Por exemplo, o que acontece quando um usuário clica em um botão específico ou como o carrinho de compras de uma loja on-line funciona.
- **Comportamento:** este aspecto está intimamente ligado à funcionalidade. Trata-se de entender o que acontece quando uma funcionalidade específica é acionada. A pergunta a ser respondida é: “como esses elementos interagem com o usuário?”. Isso poderia envolver a explicação de como um pop-up aparece quando um usuário clica em um botão específico ou como um menu se expande ou retrai ao passar o mouse.

Cada um desses aspectos é parte integrante na construção de um wireframe eficaz. Eles garantem que o design esteja centrado no usuário, tornando o website ou aplicativo fácil de usar e proporcionando uma experiência intuitiva e agradável para o usuário.

2.3 O que não incluir em um wireframe?

Um wireframe representa a estrutura fundamental de um aplicativo ou site, focando em elementos essenciais como botões, formulários e títulos. Ele não inclui elementos gráficos avançados, cores, tipografia diferenciada e deve ser apresentado em tons de cinza. O foco principal de um wireframe está na funcionalidade, comportamento, estrutura e hierarquia, não nos detalhes visuais.

Ao criar wireframes, evite usar conteúdo sem sentido, como o famoso *Lorem ipsum*. Se não tiver um conteúdo adequado ainda, use linhas horizontais para simbolizar o texto. O wireframe tem o objetivo de mostrar como o seu aplicativo ou site irá se comportar, e não como será a estética final.



Embora os wireframes sejam fundamentais para definir a estrutura, o layout e a funcionalidade de uma página web ou aplicativo, existem elementos que não são normalmente incluídos em um wireframe:

- Cores: o wireframe foca na estrutura e no layout da página, não nos detalhes de cores. As decisões de cor são normalmente tomadas durante o estágio de design de interface do usuário (UI).
- Gráficos e imagens de alta fidelidade: as imagens e gráficos podem ser indicados como espaços reservados, mas os detalhes e os ativos finais geralmente não são incluídos no wireframe.
- Fontes e tipografia: assim como as cores, as decisões sobre a fonte são geralmente tomadas mais tarde, durante o estágio de design da UI. No wireframe, a tipografia é geralmente representada por texto simples.
- Logotipos e marcas: o wireframe é focado na estrutura e funcionalidade, não na identidade da marca.
- Detalhes visuais: wireframes não são destinados a ser representações visuais detalhadas de um design. Eles fornecem uma visão geral da estrutura e do fluxo de um design. Detalhes visuais, como cores, fontes e imagens, devem ser adicionados no estágio de design.
- Comportamento: wireframes não incluem detalhes sobre o comportamento de um design. Isso inclui coisas como animações, transições e interações do usuário. O comportamento de um design deve ser especificado no estágio de design de interação.
- Conteúdo: wireframes não incluem conteúdo real, como texto, imagens ou vídeos. O conteúdo de um design deve ser adicionado no estágio de design de conteúdo.
- Especificações técnicas: wireframes não devem incluir especificações técnicas, como o tipo de código ou biblioteca a ser usada para implementar um design. A especificação técnica de um design deve ser gerada no estágio de desenvolvimento.
- Animações e transições interativas: detalhes como esses são geralmente deixados para protótipos mais avançados ou especificações de design de interface.

O objetivo de um wireframe é ser uma representação simples da página ou aplicativo, permitindo que os designers se concentrem nos aspectos



fundamentais antes de aprofundar nos detalhes do design. Isso ajuda a garantir que a estrutura e a funcionalidade sejam sólidas antes de se avançar para os aspectos mais estéticos e interativos do design.

TEMA 3 – SERVICE BLUEPRINTS

service blueprints têm como principal objetivo fornecer uma análise abrangente de como os processos internos de uma empresa influenciam a experiência do usuário final. O principal foco está em:

1. Melhorar diretamente a experiência dos colaboradores;
2. Indiretamente aprimorar a experiência do usuário final.

Um service blueprint é uma representação visual que mostra o caminho que um serviço percorre até ser entregue ao cliente. Esse mapa detalhado ilustra os diferentes pontos de contato entre o cliente e o serviço, destacando as funções e responsabilidades de cada parte envolvida.

Essa ferramenta é extremamente valiosa para aprimorar a experiência do cliente, pois ajuda a identificar áreas onde o serviço pode ser aperfeiçoado e assegura que todas as partes envolvidas na prestação do serviço estejam alinhadas.

Os service blueprints são empregados por uma variedade de organizações, incluindo empresas comerciais, governos e organizações sem fins lucrativos. Sua função é potencializar a experiência do cliente e garantir a entrega de serviços de alta qualidade.

service blueprint é um mapeamento detalhado que retrata as relações entre os elementos, processos e pessoas que compõem a jornada do cliente ou usuário dentro de um serviço.

3.1 Para que serve um service blueprint?

Um service blueprint serve para uma variedade de propósitos críticos em qualquer organização que preste serviços a clientes ou usuários. A principal finalidade dessa ferramenta é identificar e abordar possíveis falhas na experiência do serviço, que podem ser categorizadas em cinco principais razões:

- Definição incorreta do que os consumidores/clientes esperam: em alguns casos, os gerentes podem ter uma compreensão equivocada das



necessidades e expectativas dos clientes. Essa desconexão pode levar a uma experiência de serviço insatisfatória, pois o serviço fornecido pode não atender às necessidades reais dos clientes.

- Definição de padrões de serviço errados: às vezes, os gerentes podem definir padrões de serviço que não estão alinhados com o que os clientes valorizam. Isso pode resultar em desperdício de recursos em áreas que não são importantes para os clientes, enquanto outras áreas que são de alta prioridade para os clientes são negligenciadas.
- Não cumprimento dos padrões de qualidade esperados pelos funcionários: mesmo quando os padrões de serviço estão corretamente definidos, os funcionários podem não conseguir atingir esses padrões devido a fatores como falta de treinamento, falta de recursos ou baixo moral.
- Expectativas equivocadas dos consumidores/clientes: isso pode ocorrer quando há uma comunicação inadequada ou enganosa da organização. Os clientes podem ter uma percepção distorcida do serviço, o que leva a uma experiência insatisfatória quando suas expectativas não são atendidas.
- A experiência atual não atende às expectativas e à jornada do cliente: às vezes, o serviço pode não ser projetado de forma a atender ao fluxo natural da jornada do cliente. Isso pode criar atritos e tornar a experiência geral do cliente insatisfatória.

Ao usar um service blueprint, as organizações podem identificar essas falhas potenciais e trabalhar para resolvê-las. Isso pode envolver a reformulação do serviço, a educação dos clientes sobre o que esperar, a melhoria da formação dos funcionários ou a alteração da forma como o serviço é comunicado aos clientes. Ao fazer isso, o service blueprint ajuda a melhorar a qualidade do serviço, a satisfação do cliente e, em última análise, o sucesso da organização.

3.2 Vantagens dos service blueprints

Os service blueprints trazem uma série de vantagens importantes para qualquer organização que busca otimizar sua entrega de serviços e aprimorar a experiência geral do cliente.



Aqui estão algumas dessas vantagens, juntamente com exemplos para ilustrá-las:

- **Visão geral do serviço:** ao criar um blueprint, obtemos uma visão abrangente de todo o serviço, o que nos permite mapear e identificar possíveis falhas na jornada do cliente. Por exemplo, se estivermos operando um restaurante, um blueprint pode nos ajudar a identificar se os pedidos estão demorando muito para chegar à mesa do cliente devido a um gargalo na cozinha.
- **Melhoria da experiência do cliente e do funcionário:** com a comparação da jornada atual do cliente com uma jornada ideal, podemos entender e abordar áreas de melhoria, não apenas para o cliente, mas também para os funcionários. Isso ajuda a alinhar todas as expectativas e a otimizar a entrega do serviço. Por exemplo, em uma empresa de call center, um blueprint pode revelar que os funcionários estão demorando muito para resolver as consultas dos clientes porque estão usando sistemas desatualizados, afetando a satisfação do cliente e do funcionário.
- **Aumento da eficiência:** ao identificar pontos de melhoria no serviço, um blueprint também pode revelar oportunidades para aumentar a eficiência operacional. Isso pode ajudar a melhorar a comunicação entre diferentes departamentos ou equipes, garantindo uma entrega de serviço mais suave e eficaz. Por exemplo, em uma clínica médica, um blueprint pode mostrar que os pacientes estão esperando muito tempo entre a triagem e a consulta porque a comunicação entre a recepção e os médicos não está eficiente.
- **Redução de custos:** além de melhorar a eficiência, um blueprint também pode revelar áreas onde recursos podem ser economizados sem comprometer a qualidade do serviço. Isso pode ajudar a empresa a gerenciar seus recursos de forma mais eficaz e a reduzir custos. Por exemplo, em uma loja de varejo, um blueprint pode indicar que o pessoal está sendo subutilizado durante certos horários do dia, sugerindo que a programação dos funcionários pode ser ajustada para economizar em custos de mão de obra.
- **Melhoria da conformidade:** por fim, um blueprint pode ajudar a garantir que o serviço esteja em conformidade com todas as regulamentações aplicáveis. Isso pode reduzir o risco de violações regulatórias e suas



consequências associadas, como multas ou danos à reputação. Por exemplo, em uma instituição financeira, um blueprint pode mostrar se todos os processos de verificação de identidade e de combate à lavagem de dinheiro estão sendo seguidos em todas as etapas da jornada do cliente.

3.3 Quais são os benefícios do service blueprint?

Os service blueprints proporcionam diversos benefícios vitais para as organizações que buscam otimizar a entrega de serviços. Essas ferramentas visuais ajudam a ilustrar os processos operacionais, destacando possíveis gargalos, redundâncias e ineficiências. Vamos explorar três dos principais benefícios do uso do service blueprint:

- **Exposição das ineficiências dos processos:** assim como uma fotografia de alta resolução revela cada detalhe de uma cena, um service blueprint detalhado fornece uma visão clara e abrangente dos processos dentro da empresa. Essa visão clara pode revelar onde e por que os atrasos, erros ou outras ineficiências estão ocorrendo. Por exemplo, se um call center está experimentando tempos de espera longos, um service blueprint pode mostrar se o problema está na distribuição das chamadas, no tempo de resposta dos atendentes ou em algum outro lugar do processo.
- **Oportunidades de otimização:** o service blueprint não apenas expõe ineficiências, mas também pode revelar redundâncias, em que tarefas ou processos estão sendo repetidos sem necessidade. Removendo essas redundâncias, é possível economizar tempo e energia dos colaboradores, melhorar a experiência do cliente e diminuir o risco de informações inconsistentes. Por exemplo, se uma loja on-line está pedindo aos clientes para inserirem suas informações de envio várias vezes durante o processo de checkout, um service blueprint pode identificar essa redundância e sugerir uma otimização.
- **Melhoria na visão geral do serviço:** por último, mas não menos importante, um service blueprint proporciona uma visão panorâmica de todo o serviço e seus processos inter-relacionados. Isso permite que todas as áreas envolvidas entendam o fluxo completo do serviço, identifiquem pontos de melhoria e colaborem de forma mais eficaz para melhorar o sistema como

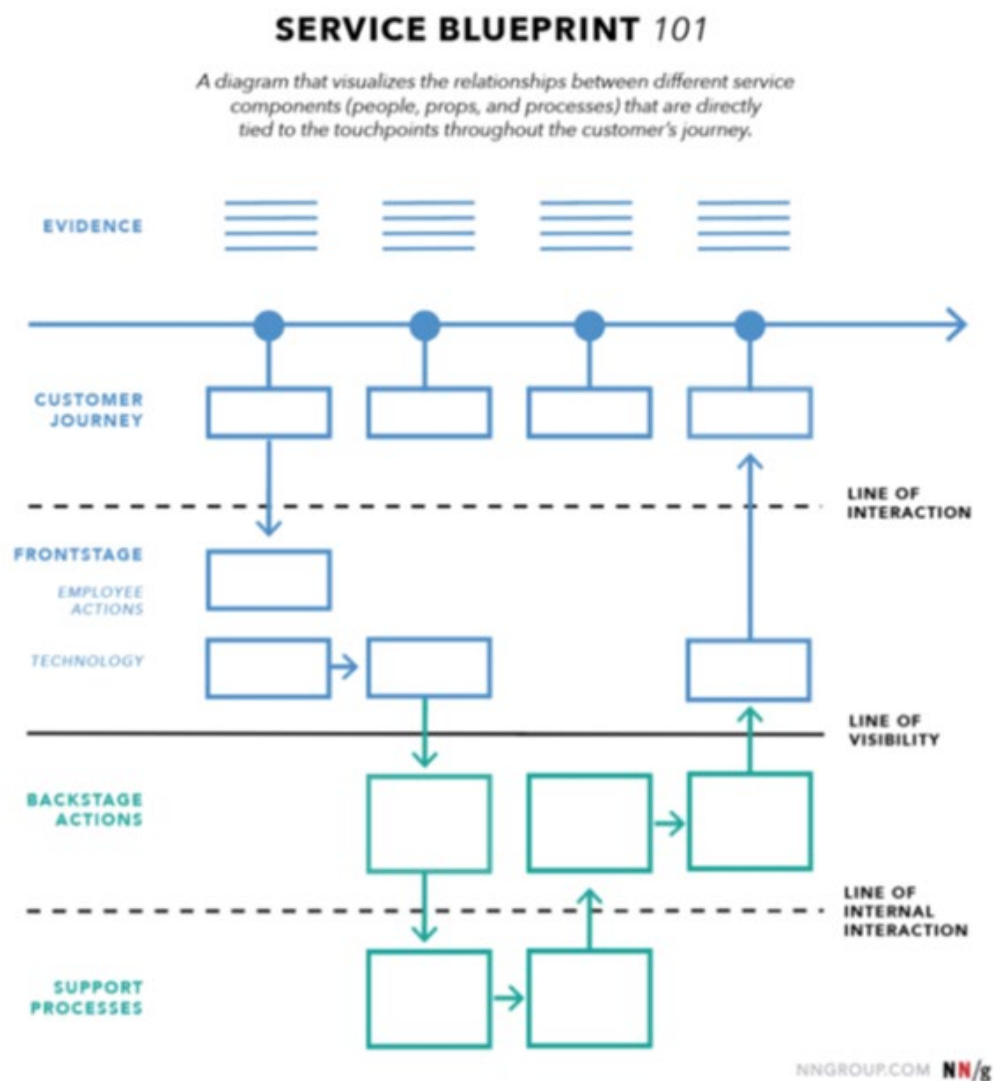


um todo. Em muitas organizações, os departamentos podem estar concentrados apenas em suas próprias atividades, perdendo a visão do quadro geral. O service blueprint ajuda a garantir que todas as partes do sistema estejam funcionando harmoniosamente para uma entrega de serviço eficiente e de alta qualidade.

3.4 Elementos essenciais

Os service blueprints são compostos por elementos fundamentais que mapeiam a jornada do usuário e o funcionamento interno de um serviço. Esses elementos essenciais são apresentados a seguir.

Figura 1 – Exemplo de diagrama de service blueprint



Fonte: NN/G.



- Ações do cliente/usuário: essas são as etapas, escolhas e interações que os clientes ou usuários realizam ao longo da jornada do serviço. Pode incluir tudo, desde abrir um aplicativo até falar com um atendente de serviço ao cliente.
- Ações de frontstage: estas são as ações que ocorrem diretamente entre o usuário final e o provedor do serviço. Elas podem ser interações humano-humano (por exemplo, um cliente conversando com um atendente de loja) ou interações humano-máquina (por exemplo, um usuário navegando em um site ou aplicativo).
- Atividades de bastidores (backstage): estas são as atividades que acontecem fora do campo de visão do usuário final, mas que são cruciais para a entrega do serviço. Isso pode incluir coisas como gestão de estoque, manutenção de sistemas de TI ou coordenação de logística.
- Processos de suporte: são as operações e interações que fornecem suporte aos colaboradores para que eles possam entregar o serviço ao usuário final. Isso pode variar desde o treinamento de funcionários até o suporte técnico interno.

O service blueprint também inclui algumas linhas importantes de separação para diferenciar entre diferentes níveis de interação e visibilidade:

- Linha da interação: representa as interações diretas entre o cliente e a organização.
- Linha da visibilidade: separa as atividades visíveis ao cliente (frontstage) das atividades invisíveis para ele (backstage). Por padrão, as atividades de frontstage ficam acima da linha e as de backstage ficam abaixo dela.
- Linha da interação interna: separa os funcionários que têm contato direto com o cliente daqueles que não têm interação direta ou oferecem suporte ao cliente.

Finalmente, o service blueprint deve incluir evidências de serviço. Estas são representações tangíveis ou intangíveis dos elementos que são utilizados nos processos e que dão suporte para que o serviço aconteça. Isso pode incluir tudo, desde uma fatura ou recibo até uma interface de usuário de um aplicativo.

As evidências de serviço podem estar tanto no frontstage (como um recibo entregue a um cliente) quanto no backstage (como um sistema de gerenciamento de pedidos usado pelos funcionários). Isso destaca o fato de que



cada aspecto de um serviço, visível ou não ao cliente, contribui para a experiência geral do serviço.

TEMA 4 – MODELO ATÔMICO

O design atômico, uma metodologia de criação de sistemas de design proposta por Brad Frost, baseia-se em componentes menores e gerenciáveis que são combinados para formar estruturas mais complexas, como mostra a Figura 2. Essa abordagem oferece uma maneira eficiente e flexível de construir sistemas de design coesos, escaláveis e consistentes.

Figura 2 – Componentes do modelo atômico

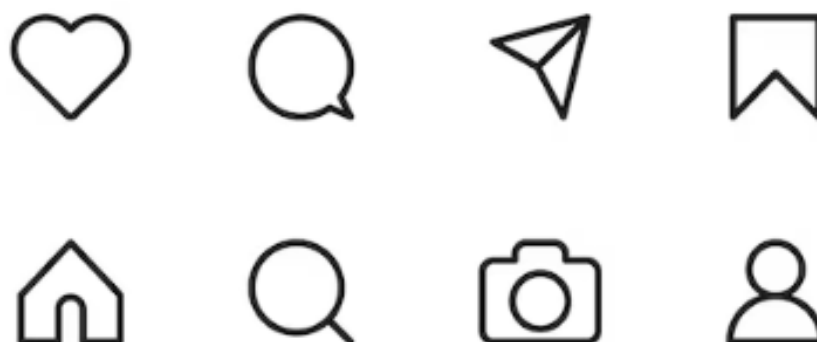


Fonte: Fung, 2023.

Para entender melhor o design atômico e sua aplicabilidade, vamos usar o Instagram como exemplo.

- **Átomos**: no contexto do design atômico, os átomos representam os elementos visuais mais básicos, como botões, ícones, campos de texto, cores e tipografia. No Instagram, por exemplo, um átomo poderia ser o botão de curtir, o ícone do Direct ou a fonte usada para os textos, exemplos apresentados na Figura 3.

Figura 3 – Exemplos de botões encontrados no Instagram

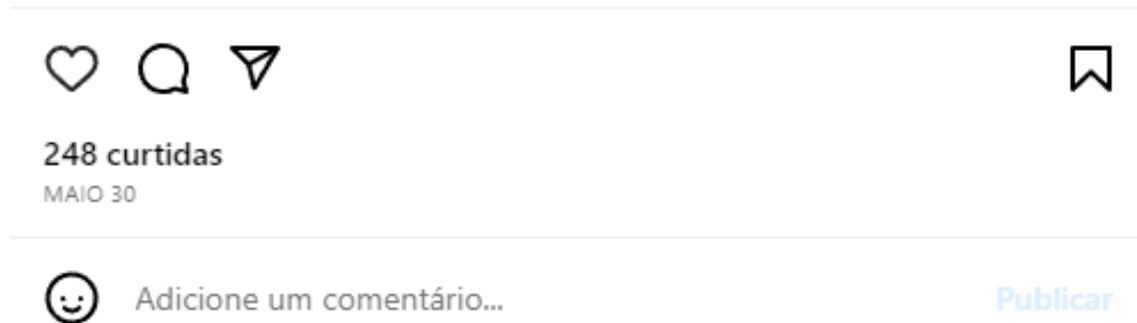


Fonte: Elaborado por Fung, 2023.



- Moléculas: as moléculas são formadas pela combinação de vários átomos para criar componentes mais complexos. A barra de navegação inferior do Instagram, que inclui vários ícones (átomos), serve como exemplo de uma molécula, conforme Figura 4.

Figura 4 – Barra inferior de navegação de um feed do Instagram

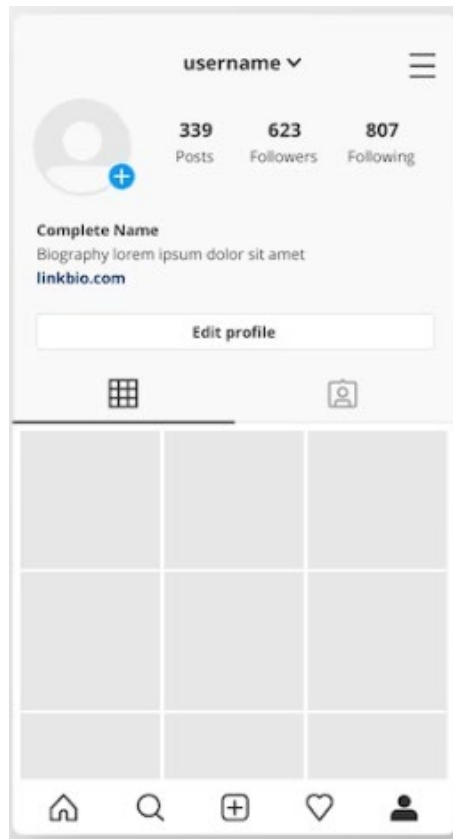


Fonte: Elaborado por Fung, 2023, com base em Instagram, [S.d.].

- Organismos: os organismos são conjuntos de moléculas que funcionam juntas para formar um componente funcional maior do design. A tela de perfil do Instagram, que inclui várias moléculas (como a foto do perfil, o nome de usuário, a barra de navegação e o feed de fotos), pode ser considerada um organismo, visualizado na Figura 5.

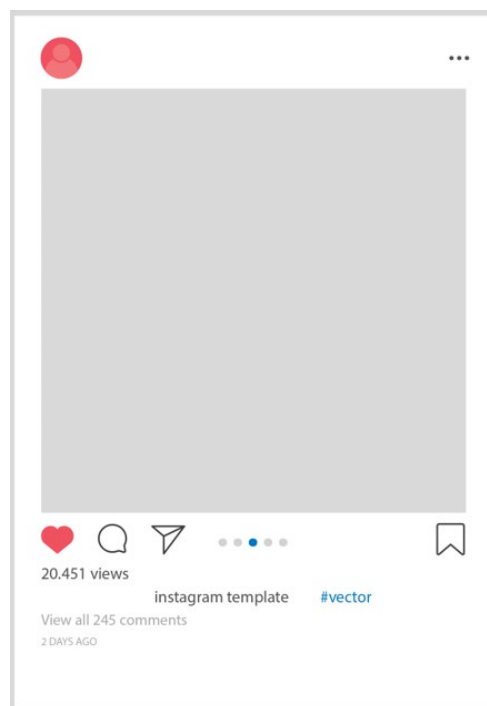


Figura 5 – Exemplo de tela de perfil do Instagram



Fonte: Elaborado por Fung, 2023, com base em Instagram, [S.d.].

Figura 6 – Exemplo de modelo de feed do Instagram

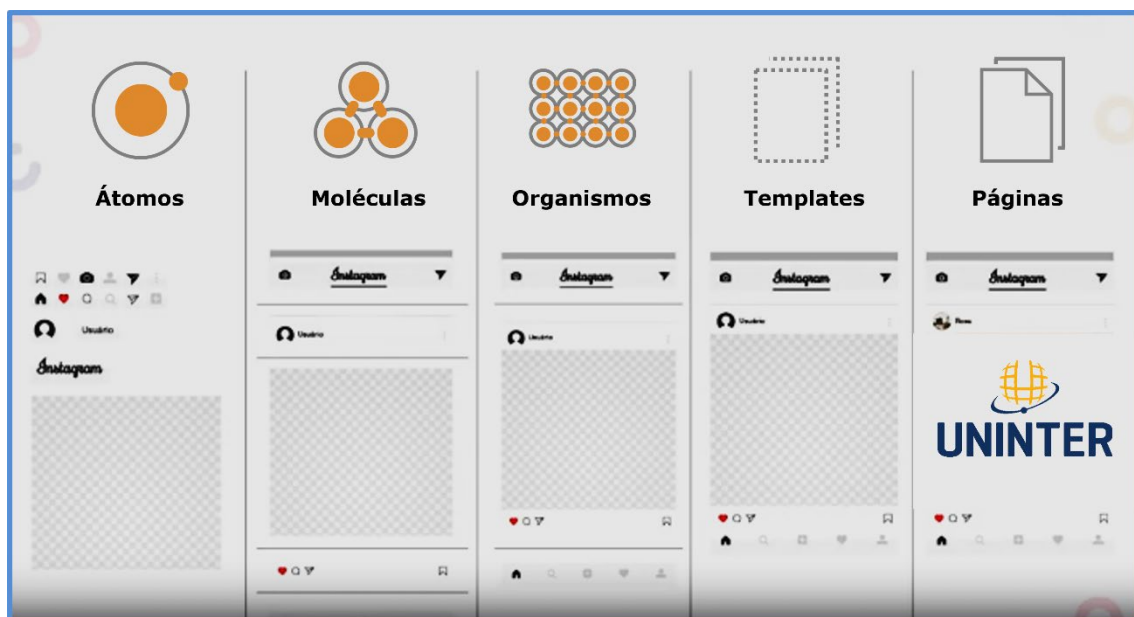


Fonte: Elaborado por Fung, 2023, com base em Instagram, [S.d.].



- Modelos (templates): modelos são esqueletos das páginas onde os organismos são organizados em um layout, formando a estrutura final do design. O modelo não contém conteúdo real, mas delinea a estrutura na qual o conteúdo será inserido. No Instagram, um modelo poderia ser a estrutura básica de uma postagem no feed, com espaços reservados para a foto, o nome de usuário, os ícones de curtir/comentar/compartilhar e a caixa de comentários, exemplificado na Figura 6.
- Páginas: as páginas são a última etapa do design atômico, onde o modelo é preenchido com conteúdo real. Por exemplo, uma postagem individual no feed do Instagram é uma página, pois contém uma foto específica, o nome de usuário da pessoa que a postou, curtidas, comentários e assim por diante.

Figura 7 – Exemplos dos componentes de um modelo atômico aplicado ao Instagram



Fonte: Fung, 2023.

O design atômico, sendo um processo iterativo, permite a revisão e o ajuste do design conforme necessário. Após definir todos os elementos, é possível começar a testar, revisar e validar o design, aprimorando e adaptando-o continuamente para atender às necessidades do projeto.

O design atômico, concebido por Brad Frost, é uma abordagem altamente eficiente e flexível que auxilia na manutenção da consistência visual e funcional



em todo o sistema de design. A capacidade de reutilizar componentes existentes economiza tempo e recursos, tornando o design atômico uma escolha popular entre designers e desenvolvedores.

4.1 Vantagens do modelo atômico

O design atômico é um sistema fundamental para impulsionar a organização em um projeto. Sua adoção conduz a equipe a estabelecer padrões para cada elemento envolvido, culminando em um design sistemático mais coeso e harmonizado.

Essa metodologia favorece uma aceleração no ritmo de desenvolvimento. O motivo reside na visão de que os elementos são interconectados e complementares. Isso facilita a realização de ajustes, mitigando a necessidade de retrabalho e economizando tempo precioso.

O design atômico também atua como uma ponte, estreitando as relações entre as equipes de desenvolvimento e design. A implementação dessa abordagem requer colaboração e comunicação eficaz, o que minimiza desentendimentos e divergências de opiniões, alinhando todos em direção a um objetivo comum.

A prototipação é outra fase que se beneficia grandemente do design atômico. Com uma definição clara dos elementos desde o início, a criação e os testes dos protótipos tornam-se tarefas menos árduas. Além disso, se a necessidade de alterações surgir durante as pesquisas, o Atomic Design permite a reorganização ou modificação dos itens de forma prática e eficiente, antes da construção do produto final.

4.2 Desvantagens do modelo atômico

Embora o design atômico ofereça benefícios substanciais, também existem desafios potenciais na sua aplicação. Um desses desafios pode ser reestruturar a equipe de acordo com o atomic design, especialmente ao integrá-lo em produtos já em desenvolvimento.

Uma questão frequentemente levantada por alguns profissionais é a discordância gerada pela falta de alinhamento. Isso pode não ser um problema intrínseco à metodologia, mas sim uma consequência da maneira como ela é implementada. Se o design atômico não for adequadamente comunicado e



documentado entre todos os membros da equipe, ele pode, paradoxalmente, gerar mais conflitos do que soluções.

Ainda nesta linha, algumas equipes podem encontrar dificuldades em classificar determinados elementos dentro dos grupos definidos pelo design atômico (átomos, moléculas, organismos, templates ou páginas). Nesse aspecto, o crucial é estabelecer a classificação de um elemento em consonância com os objetivos da equipe e garantir que todos estejam a par dessa decisão.

O design atômico é um processo que precede o desenvolvimento. Em projetos menos complexos ou com prazos apertados, a implementação desta metodologia pode ser contraproducente e acabar por retardar a criação do produto. Portanto, é imprescindível estar em sintonia com todas as partes envolvidas para avaliar a adequação e viabilidade do design atômico para cada projeto.

Saiba mais

Curiosidades sobre Brad Frost

Brad Frost é um respeitado designer de UX e autor baseado em Chicago. É mais conhecido como o criador do atomic design, uma metodologia de design de sistemas que organiza os elementos visuais de um sistema em blocos de construção menores e mais gerenciáveis. Essa abordagem ao design tem sido influente na comunidade de UX, oferecendo uma maneira estruturada e eficiente de criar interfaces de usuário consistentes e reutilizáveis.

Antes de se tornar uma figura central no campo do design de sistemas, Frost trabalhou como designer web e desenvolvedor front-end em várias agências digitais. Suas contribuições ao design responsivo e ao design de sistemas são profundas e ele é conhecido por sua abordagem cuidadosa e minuciosa.

Como um frequente palestrante em conferências e eventos, Frost compartilha suas ideias e metodologias com a comunidade mais ampla de design. Ele é o autor de vários livros, incluindo *Atomic Design* e *Designing interfaces*, que são referências no campo do design de UX.

Frost também é o fundador de um site próprio², no qual disponibiliza artigos, tutoriais e outros recursos sobre design de UX. Ele é coautor do livro *The*

² Disponível em: <<https://bradfrost.com>>. Acesso em: 2 ago. 2023.



web standards creativity, uma compilação de técnicas inovadoras para designers e desenvolvedores web. Além disso, Frost é o autor de *This is responsive*, um recurso on-line que fornece padrões, recursos e notícias sobre design responsivo.

Brad Frost é um membro ativo da comunidade de design, e seu trabalho foi reconhecido por organizações como AIGA, Adobe e Nielsen Norman Group. Ele recebeu vários prêmios por seu trabalho, que tem sido destaque em publicações como Wired, Fast Company e The New York Times.

Brad Frost é um designer talentoso e experiente que fez contribuições significativas para o campo do design de UX. Ele continua a ser um recurso valioso para designers e desenvolvedores ao redor do mundo, e seu trabalho na criação do atomic design está ajudando a melhorar a experiência do usuário para todos.

TEMA 5 – DESIGN SPRINT

O design sprint é um processo inovador desenvolvido por Jake Knapp e a equipe do Google Ventures, que permite a empresas, startups e equipes solucionarem problemas complexos e desenvolverem novas ideias em um curto período de tempo — geralmente cinco dias, também conhecido como *sprint*. O objetivo desse método é projetar, prototipar e testar ideias de produtos ou serviços de maneira rápida e eficaz.

Figura 8 – Atividades desenvolvidas por dia do design sprint



Fonte: Medium, 2017.

Algumas das vantagens do design sprint:

- ajuda as equipes a resolverem problemas rapidamente e eficientemente;
- gera ideias inovadoras e eficientes para novos produtos e serviços;
- é envolvente e incentiva a colaboração;



- pode ser usado por equipes de todos os tamanhos.

O processo de design sprint é estruturado em cinco etapas que ocorrem durante uma semana:

1. Entendimento ou mapeamento e escolha do alvo: no primeiro dia, a equipe se dedica a entender o problema em mãos, mapear o processo ou o produto e escolher um elemento específico para focar. Aqui, é importante definir o desafio que será enfrentado e compreender o objetivo a longo prazo. “O que queremos no final deste sprint?” e “se lançarmos este produto, quais são os possíveis problemas?” são perguntas essenciais a serem respondidas. A equipe cria um mapa da jornada do usuário, com a ajuda de especialistas da área, transformando o problema em uma oportunidade.
2. Remix ou esboço: no segundo dia, são criadas soluções potenciais para o problema escolhido. Cada membro da equipe desenha soluções individualmente para garantir uma variedade de ideias. Esse é o momento da ideação, em que a equipe busca referências de mercado e utiliza técnicas de brainstorming, como o Crazy 8s, para gerar ideias inovadoras.
3. Decisão: no terceiro dia, decide-se qual dessas soluções será prototipada. As soluções são avaliadas e votadas pela equipe. Aqui, todos os sketches das soluções são apresentados e os pontos mais interessantes ou o sketch inteiro mais votado é escolhido para prosseguir. É o decisor quem faz a escolha final.
4. Protótipo: o quarto dia é dedicado à transformação da solução escolhida em um protótipo realista. A equipe desenvolve um storyboard detalhado com fluxos de usuário e desenha todas as telas da solução.
5. Teste de usabilidade: no último dia, o protótipo é testado com usuários reais. A equipe observa como os usuários interagem com o protótipo para aprender mais sobre o que funciona e o que não funciona. Essa etapa é dividida em duas: uma entrevista inicial para entender o usuário e seu contexto, seguida do teste do protótipo em si. Todo o processo de teste é compartilhado com a equipe para que todos possam analisar as interações dos usuários.

O design sprint é eficaz porque permite que as equipes aprendam rapidamente, em vez de gastarem semanas ou meses desenvolvendo um



produto ou serviço completo, apenas para descobrir que ele não atende às necessidades dos usuários. Entre suas vantagens, destacam-se a resolução rápida de problemas, a geração de ideias inovadoras, o incentivo à colaboração e a possibilidade de testar soluções com usuários e obter feedback em tempo real.

Para realizar um design sprint, é necessário ter um desafio significativo, uma equipe de colaboradores multidisciplinar, um decisor entre os colaboradores da equipe e materiais para a equipe, como flipcharts, lousas, post-its, canetas e papel. Além disso, é importante lembrar que o design sprint é um processo iterativo. Ele pode ser repetido várias vezes até que todos os ajustes necessários sejam feitos, permitindo que ideias sejam refinadas e testadas com baixo custo.

Um exemplo real do uso do design sprint é o da Slack³, uma ferramenta de comunicação empresarial. Em 2015, a equipe da Slack estava lutando para desenvolver uma nova funcionalidade chamada *Post*. Eles sabiam que queriam criar um meio para os usuários compartilharem mensagens longas, mas não tinham certeza de como essa funcionalidade deveria funcionar. Eles decidiram usar um design sprint para resolver o problema. Ao final do sprint, eles tinham um protótipo funcional da funcionalidade *Post* que foi capaz de testar com os usuários. O feedback dos usuários foi incorporado na versão final do *Post*, que é uma das funcionalidades mais populares da Slack.

A metodologia design sprint é poderosa e permite às equipes desenvolverem e validarem ideias rapidamente, minimizando os riscos e o tempo de desenvolvimento. Ela estimula a colaboração, a inovação e a resolução eficiente de problemas, tornando-se uma ferramenta valiosa para equipes de todos os tamanhos.

5.1 Desenvolvendo o design sprint

O design sprint é uma metodologia ágil de cinco dias, estrategicamente estruturada, que orienta equipes através de uma jornada de inovação para resolver desafios significativos. Cada dia da semana é cuidadosamente planejado para cumprir uma etapa específica do processo. Aqui está uma visão geral de como essa semana irá se desenvolver:

³ Disponível em: <<https://slack.com>>. Acesso em: 2 ago. 2023.



- **1º dia: desenvolvendo o design sprint**

O primeiro dia do design sprint é dedicado à compreensão do problema, ao entendimento do negócio e ao nivelamento do conhecimento entre todos os participantes. O objetivo a longo prazo é definido através de perguntas como “o que queremos alcançar ao final deste sprint?” e “se lançarmos este produto, quais são os possíveis desafios?”.

A jornada do usuário é mapeada e, com esse mapa, a equipe consegue visualizar os passos que o usuário percorre para atingir seu objetivo. A fase “Ask the Experts” segue, em que especialistas na área são convidados para entrevistas, trazendo perspectivas e conhecimentos valiosos.

A análise destas entrevistas, juntamente com a revisão do mapa, permite à equipe transformar o problema em uma oportunidade. Com o desafio compreendido, inicia-se a fase “How Might We” (Como podemos), em que todos os possíveis caminhos para resolver o problema são levantados e votados. Neste ponto, o voto do especialista tem mais peso, ajudando a definir as principais estratégias.

Os “How Might We” são anotados em post-its e colocados no mapa, indicando os pontos-chave do sprint. Ao final do dia, o foco é definido do ponto de vista do usuário, assim como os participantes que realizarão o teste de usabilidade.

- **2º dia: desenvolvendo o design sprint**

No segundo dia, denominado *Remix*, a equipe busca referências de mercado que possam ajudar a solucionar o problema. Esse é o dia da ideação, em que a equipe faz brainstorming e utiliza a técnica Crazy 8s para gerar ideias – não necessariamente soluções – que podem contribuir para resolver o problema.

- **3º dia: desenvolvendo o design sprint**

No terceiro dia, os esboços (sketches) desenvolvidos no dia anterior são expostos para votação. A equipe pode votar na história inteira ou em partes específicas das ideias. As ideias mais votadas são então transformadas em um storyboard, uma história visual da solução proposta.

O decisor (geralmente o stakeholder ou o proprietário do produto) analisa os votos e decide qual(uais) ideia(s) seguir. A partir daí, a equipe desenvolve o



storyboard, que detalha o fluxo do usuário e as telas individuais da solução, garantindo que a ideia é consistente e não possui lacunas.

- **4º dia: desenvolvendo o design sprint**

O quarto dia é dedicado à criação do protótipo de alta fidelidade. Todas as ideias e soluções discutidas e aprovadas nos dias anteriores são integradas nesse protótipo. O cliente pode participar neste dia, contribuindo para o desenvolvimento do protótipo. Paralelamente, a equipe realiza o recrutamento dos usuários para o teste de usabilidade e desenvolve o questionário que será utilizado para testar as hipóteses.

- **5º dia: desenvolvendo o design sprint**

O último dia do design sprint é dedicado ao teste de usabilidade. O teste é dividido em duas partes: uma entrevista contextual para entender a rotina e o contexto do usuário, e o teste do protótipo. É importante esclarecer para os usuários que eles testarão um protótipo, e não um produto final.

Os resultados do teste são compartilhados com toda a equipe para análise. Se houver um erro que ocorra com todos os testadores, é recomendado fazer a correção e refazer o teste com todos novamente, garantindo a validade dos resultados.

Lembre-se: o design sprint é um processo flexível e essas etapas podem variar dependendo das necessidades específicas do projeto ou da equipe.

Exemplo de um design sprint de cinco dias aplicado ao desenvolvimento de um aplicativo de delivery de comida

- **1º dia: entendimento e definição do desafio**

O problema identificado é a dificuldade dos usuários em encontrar opções de comida saudável na área local deles. Após a discussão em grupo, definimos o objetivo a longo prazo: “Criar um aplicativo que facilite o encontro de opções de comida saudável local”. Ouvimos especialistas em alimentação saudável, nutrição e delivery de alimentos. Depois de mapear a jornada do usuário e revisá-la com base nas entrevistas dos especialistas, identificamos oportunidades e desafios.

- **2º dia: esboçando soluções**



No segundo dia, realizamos uma sessão de brainstorming para gerar ideias. Referências de aplicativos similares e técnicas como Crazy 8s foram utilizadas para criar uma variedade de soluções potenciais. Os esboços de ideias incluíram opções de filtragem personalizadas, programas de recompensa de saúde e parcerias com nutricionistas e restaurantes locais.

• 3º dia: decisão e storyboard

Revisamos todos os esboços e votamos nas melhores ideias. Decidimos focar em um sistema de filtragem robusto que permite aos usuários encontrar refeições saudáveis baseadas em suas preferências dietéticas, e um programa de recompensas que oferece incentivos para escolhas saudáveis. Depois de decidir o caminho a seguir, criamos um storyboard detalhado mostrando como o usuário interagiria com essas características.

• 4º dia: prototipação

Utilizando ferramentas de design digital, criamos um protótipo de alta fidelidade do aplicativo, com base no storyboard. O protótipo incluiu elementos-chave como a tela inicial, a interface de busca, as páginas de refeições e o sistema de recompensas. Recrutamos usuários-alvo para testar o protótipo no dia seguinte.

• 5º dia: teste de usabilidade

Os usuários participaram de uma sessão de teste de usabilidade, durante a qual interagiram com o protótipo. Pedimos que completassem tarefas específicas, como encontrar uma refeição vegetariana na área local ou resgatar pontos de recompensa. Observamos suas interações e recolhemos feedback sobre a usabilidade e funcionalidade do aplicativo. Ao final do dia, tínhamos informações valiosas para informar a próxima iteração do design.

5.2 Resultados

O design sprint é uma poderosa metodologia que permite validar uma ideia diretamente com os usuários finais em um curto período de tempo. Com a utilização de protótipos, é possível experimentar, avaliar e aperfeiçoar ideias, gerando novas versões para testes de maneira eficiente e econômica, em comparação com o desenvolvimento integral de um produto.



Esse processo é iterativo, o que significa que pode ser repetido várias vezes, até que todos os ajustes necessários tenham sido feitos e uma solução satisfatória tenha sido encontrada. Essa iteração permite que as equipes aprendam rapidamente e integrem feedback valioso dos usuários, promovendo melhorias constantes no design do produto.

Além disso, o design sprint promove a colaboração e o alinhamento entre diferentes partes interessadas, desde designers e desenvolvedores até executivos e usuários. Ele desmistifica o processo de tomada de decisão no design, tornando-o transparente e participativo. Ao final de um design sprint, as equipes não só têm um protótipo testado, mas também um melhor entendimento das necessidades do usuário e uma direção clara para o desenvolvimento futuro do produto.

Deve-se notar, entretanto, que o design sprint não é uma solução definitiva para todos os problemas. Ele é mais eficaz quando usado para responder a perguntas críticas de negócios e reduzir incertezas, oferecendo um ponto de partida sólido para o desenvolvimento de um produto ou serviço.

Saiba mais

Curiosidades sobre Jake Knapp

Jake Knapp é o criador do design sprint, um processo de cinco dias para resolver problemas e testar novas ideias rapidamente. Antes de criar o design sprint, Knapp passou dez anos trabalhando no Google, onde atuou em produtos como o Gmail e o Google Hangouts.

Enquanto estava no Google, Knapp percebeu que as equipes de projeto frequentemente gastavam muito tempo em reuniões de planejamento e discussão sem chegar a um produto final. Para resolver esse problema, ele começou a experimentar diferentes formas de agilizar o processo de desenvolvimento de produtos. Isso o levou à criação do design sprint, um processo estruturado de cinco dias que combina métodos de estratégia de negócios, inovação, ciência do comportamento e design thinking.

Knapp desenvolveu e refinou o processo do design sprint enquanto trabalhava no Google Ventures, o braço de capital de risco do Google. No Google Ventures, ele e sua equipe realizaram mais de 150 sprints com empresas do portfólio, como Nest, Slack, 23andMe e Uber, entre outros.

• Algumas curiosidades sobre o design sprint



O design sprint foi inspirado por muitos métodos de trabalho diferentes, incluindo o design thinking de Stanford, a estratégia de negócios de Harvard e o movimento de inicialização enxuta de Eric Ries.

O livro de Knapp, *Sprint: how to solve big problems and test new ideas in just five days*, é um guia detalhado sobre como conduzir um design sprint. Foi publicado em 2016 e tem sido usado como uma referência valiosa por muitas organizações.

Embora o processo original do design sprint seja de cinco dias, Knapp e sua equipe também desenvolveram uma versão condensada de dois dias para equipes com restrições de tempo.

O design sprint se tornou tão popular que agora é usado por muitas empresas fora do ecossistema do Google, incluindo Lego, McKinsey e a BBC.

O próprio Jake Knapp descreve o design sprint como “o maior truque” em seu livro, pois permite que as empresas evitem cometer erros caros e demorados ao testar uma ideia ou produto em um ambiente seguro e controlado.

FINALIZANDO

Ao longo deste estudo, exploramos os aspectos cruciais do design de experiência do usuário (UX), design de interface do usuário (UI), prototipação, diagramas de serviços (service blueprints), modelo atômico e design sprint. Cada um desses elementos contribui de maneira única para a criação de produtos digitais bem-sucedidos e experiências de usuário memoráveis.

Por meio do design de UX e UI, nós nos esforçamos para compreender as necessidades, desejos e comportamentos dos usuários, criando interfaces intuitivas e atraentes que facilitam a interação entre o usuário e o produto digital. A prototipação e o design de serviço nos permitem mapear a jornada do usuário e visualizar a interação de todos os componentes do sistema, revelando pontos de contato críticos e áreas de melhoria.

Ao implementar o modelo atômico, introduzimos uma estrutura escalável e flexível para o design, que considera cada elemento de interface como parte de um todo coeso, permitindo uma manutenção mais eficiente e design consistente.

Finalmente, por meio do design sprint, tivemos uma visão de como condensar o processo de design em um período intensivo, porém produtivo, que permite validar ideias diretamente com os usuários finais. Essa abordagem



rápida e iterativa destaca a importância do aprendizado contínuo, do feedback do usuário e da melhoria constante no ciclo de vida do design de produtos.

Ao combinar esses elementos, somos capazes de criar produtos digitais que não apenas atendem às necessidades funcionais dos usuários, mas também proporcionam experiências de uso enriquecedoras e agradáveis. Lembre-se: o design não é apenas uma questão de estética, mas também de função e experiência. E, no centro de tudo, estão os usuários, com suas necessidades, suas expectativas e suas experiências.



REFERÊNCIAS

BAXTER, M. **Projeto de produto**: guia prático para o design de novos produtos. São Paulo: Edgard Blucher, 2008.

CHANDLER, C.; UNGER, R. **O guia para projetar UX**: a experiência do usuário (UX) para projetistas de conteúdo digital, aplicações e websites. 1. ed. Starlin Alta Consult, 2009.

GARRET, J. J. **The Elements of user experience**: user centered design for the web and beyond. New Readers Press, 2011

JOHNSON, S. **Cultura da interface**: como o computador transforma nossa maneira de criar e comunicar. Rio de Janeiro: Zahar, 2001.

NORMAN, D. **Emotional design**: why we love (or hate) everyday things. New York: Basic Books, 2004.

_____. **The design of everyday things**. New York, Basic Books, 1988.

PREECE, J.; ROGER, Y.; SHARP, H. **Design de interação**. 3. ed. Porto Alegre: Bookman Companhia ED, 2013.

ROGERS, Y.; SHARP, H.; PREECE, J. **Design de interação**: além da interação homem-computador. 3. ed. Bookman, 2013.

SANTOS, R. Usabilidade de interfaces e arquitetura da informação: alguns aspectos da organização de conteúdo para o meio digital. In: **Anais Abergo**, Gramado, 2000.

SOUSA, J. O. design sprint (Resumo). **Medium**, 7 fev. 2017. Disponível em: <<https://medium.com/gobee/design-sprint-resumo-3a5b634bd311>>. Acesso em: 2 ago. 2023.