DESENVOLVIMENTO DE SOFTWARE COM METODOLOGIAS ÁGEIS

Rafael Gastão Coimbra Ferreira

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS

Métodos ágeis, *design* de interação e *user experience* (UX)

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Identificar a importância do *design* de interação no desenvolvimento de *software*.
- Explicar o papel da *user experience* (UX) em projetos de *software*.
- Empregar técnicas de *design* de interação e UX no desenvolvimento ágil.

Introdução

A participação de usuários tem se tornado cada vez mais importante na criação de um *software*. A finalidade é descobrir como eles interagem com a interface para identificar pontos negativos e positivos dessa interação, e, por consequência, otimizar sua experiência de uso antes que o programa seja efetivamente lançado. Para isso, porém, é preciso saber como identificar as necessidades do usuário-alvo e desenvolver um *design* a elas adequado. Além disso, a dinâmica de mercada demanda um desenvolvimento de *software* mais leve, ágil, iterativo e incremental, sem abdicar da qualidade, é claro, respeitando aspectos como usabilidade, acessibilidade e uma boa experiência de usuário, muito conhecida como *user experience* (UX).

Neste capítulo, você vai descobrir qual é o impacto do *design* de interação e da experiência do usuário no desenvolvimento de *software*, e como aplicar os princípios relativos a esses conceitos ao gerenciamento ágil de projetos.

1 *Design* de interação no desenvolvimento de *software*

O ***design* de interação** pode ser definido como o projeto de produtos interativos, como *sites*, objetos domésticos, *smartphones* e *software*, que buscam apoiar o modo como as pessoas se comunicam e interagem em suas vidas cotidianas, seja em casa ou no trabalho. Em outras palavras, significa criar experiências de usuário que melhorem e ampliem a maneira como as pessoas trabalham, comunicam-se ou interagem (ROGERS; SHARP; PREECE, 2013). A Figura 1 apresenta o *design* de interação como ponto central e como ele interage com outras áreas do conhecimento, como a interação humano-computador (IHC).

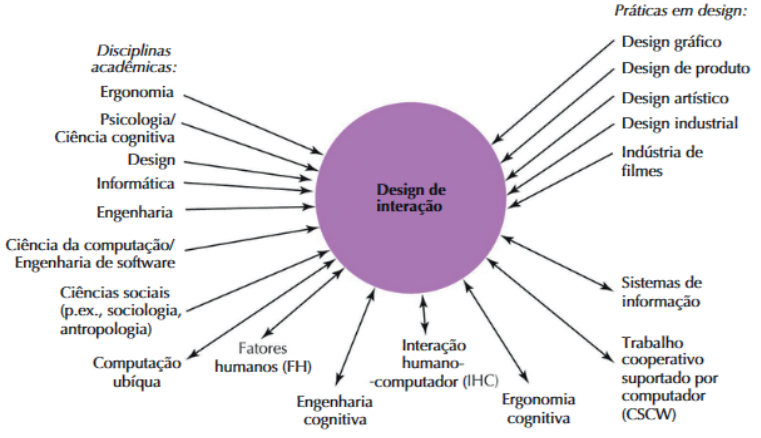

Figura 1. Campos interdisciplinares que contemplam o *design* de interação.
Fonte: Rogers, Sharp e Preece (2013, p. 10).

No desenvolvimento de *software*, o *design* de interação não é mais visto, unicamente, como dependente das características da interface do usuário, mas como o modo como o sistema se adéqua no contexto de uso. Para isso, é necessário entender as necessidades dos usuários como o meio de comunicar o processo de *design* (BERNARDINO, 2005).

Por exemplo, os engenheiros da Apple, empresa que dispensa apresentações, priorizam o desenvolvimento focado, principalmente, no usuário. Steve Jobs, seu fundador, enfatizava a importância de um bom *design* de interação, destacando que *design* não é apenas a aparência e a sensação de *design*, mas funcionalidade. Ou seja, criar e implementar uma tecnologia não é suficiente:

ela precisa existir em função do usuário final, de uma experiência coerente e satisfatória. Por isso a empresa foca em desenvolvimento de *software* fáceis de usar e intuitivos, norteado pelas necessidades do cliente (VILACA, 2017).

Segundo Cybis, Betiol e Faust (2017), existe um conjunto de qualidades que um *software* deve apresentar para atender às necessidades do usuário, otimizando sua experiência interativa. Veja-as abaixo.

- **Heurísticas de Nielsen:** trata-se de 10 heurísticas, ou seja, princípios de usabilidade, criadas por Jakob Nielsen para avaliar elementos da interface de usuário. São regras gerais, não diretrizes de usabilidade específicas. Incluem, por exemplo, correspondência entre sistema e mundo real, controle do usuário e liberdade, consistência e padrões, prevenção de erros, estética e *design* minimalistas, etc. (ROGERS; SHARP; PREECE, 2013).
- **Regras de ouro de Shneiderman:** utilizadas para guiar a criação de interfaces de usuário com boa experiência, produtivas e que não causem frustrações em seus usuários. As gigantes Apple, Google e Microsoft são algumas empresas de grande sucesso cujos produtos de *software* refletem as regras de ouro de Shneiderman. Incluem, por exemplo, atendimento à usabilidade universal, esforço pela consistência, inclusão de diálogos que indiquem o fim de uma ação, etc.
- **Princípios de *design* do Android:** são diretrizes para os padrões visuais e de navegação para plataformas Android, incluindo critérios de qualidade para compatibilidade, desempenho, segurança.
- **Princípios de diálogo da ABNT NBR ISSO 9241-110:2012:** estabelecem princípios ergonômicos de projeto como referência para a aplicação desses princípios na análise, no projeto e na validação de sistemas interativos. É destinada a projetistas de *software*, desenvolvedores, testadores e clientes, para analisar a qualidade do produto (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2012).
- **Critérios ergonômicos de Christian Bastien e Dominique Scapin:** Apresenta uma pesquisa conduzida para o design de critérios ergonômicos aplicados à avaliação de IHC. Consiste em uma lista de 18 critérios elementares, incluindo nove critérios principais, que são apresentados junto com suas definições, seus fundamentos, exemplos de diretrizes e comentários estabelecendo as distinções entre si. Esse conjunto de critérios, princípios, regras e heurísticas para proporcionar uma boa experiência de usuário e garantir sua ergonomia (Quadro 1).

Quadro 1. Critérios ergonômicos de Bastien e Scapin

Critério	Descrição	Subprincípios
Poder marcar a experiência	Menciona que a interação deve ocorrer para poder surpreender (superação das expectativas), encantar (beleza estética) e tornar simples a vida de seus usuários.	■ Poder encantar ■ Poder surpreender ■ Poder simplificar a vida
Qualidade da ajuda	Relaciona-se ao apoio que uma interface dá ao usuário em seu aprendizado do sistema e quando necessita de auxílio para concluir ou realizar uma atividade.	■ Qualidade da documentação de ajuda ■ Adequação ao aprendizado
Condução das ações do usuário	Aplica-se quando o objetivo é favorecer a utilização do sistema por usuários novatos, principalmente.	■ Apresentação do estado do sistema ■ Convite ■ *Feedback* imediato
Qualidade das apresentações	Relaciona-se ao entendimento dos elementos da tela, da legibilidade e da distribuição das informações e dos objetos.	■ Significado dos códigos e das denominações ■ Legibilidade ■ Agrupamento por distinção e localização ■ Agrupamento por distinção e formato
Carga de trabalho	Preconiza a redução de carga cognitiva e perceptiva dos usuários. Também é relacionado a um diálogo eficiente.	■ Brevidade das entradas individuais ■ Concisão das entradas individuais ■ Ações mínimas ■ Densidade informacional
Controle do usuário	Aplica-se nas tarefas longas e sequenciais, em que os processos sejam demorados, visando à otimização de tempo e à não perda de dados.	■ Ações explícitas ■ Controle do usuário

(*Continua*)

(*Continuação*)

Quadro 1. Critérios ergonômicos de Bastien e Scapin

Critério	Descrição	Subprincípios
Adaptabilidade	Estabelece como qualidade de um *software* adaptar-se ou estar adaptado a usuários com diferentes perfis e níveis de competência. Ou seja, deve servir para diferentes públicos-alvo.	■ Flexibilidade ■ Personalização ■ Consideração da experiencia do usuário
Gestão de erros	Diz respeito a todos os mecanismos que permitem reduzir ou evitar a ocorrência de erros, bem como que favoreçam sua correção.	■ Proteção de erros ■ Tolerância a erros ■ Qualidade das mensagens de erros ■ Correção de erros
Homogeneidade	Estabelece que as interfaces devem ser consistentes, procurando respeitar padrões e estilos definidos tanto no nível do produto quanto da plataforma.	■ Coerência interna a uma aplicação ■ Coerência externa a uma plataforma
Compatibilidade	Preconiza que deve existir compatibilidade com os usuários, a interface e as tarefas que são realizadas no sistema.	■ Compatibilidade com o usuário ■ Compatibilidade com as tarefas do usuário

Fonte: Adaptado de Cybis *et al.* (2017).

O *design* interativo bem-sucedido utiliza esses princípios para criar um *software* que proporcione as experiências de usuário desejadas pelos clientes. No desenvolvimento ágil, o *design* de interação deve ser contemplado, de forma que, frequentemente, ambos estão envolvidos no desenvolvimento do mesmo *software*. Todavia, o *design* de interação e o desenvolvimento ágil têm diferentes perspectivas sobre o *software*. Considerando que o *design* de interação foco em como os usuários finais trabalharão com o *software*, o desenvolvimento ágil foca em como o *software* deve ser construído.

Além disso, um dos problemas de integração dessas duas metodologias é que, tradicionalmente, elas usam abordagens diferentes sobre como os recursos devem ser alocados em um projeto. Por exemplo, as metodologias ágeis buscam

fornecer pequenos conjuntos de recursos de *software* aos clientes o mais breve possível em iterações curtas, enquanto o *design* de interação despende de que sejam realizadas pesquisas e análise antes do início do desenvolvimento com foco no usuário. Entretanto, os dois processos têm em comum uma apreciação pela importância da avaliação da satisfação do usuário e do cliente e preconizam que uma abordagem iterativa é a melhor maneira de realizar a construção de um *software* (FERREIRA; NOBLE; BIDDLE, 2007).

2 A experiência de usuário no desenvolvimento de *software*

A **experiência de usuário** engloba todos os aspectos da interação que as pessoas costumam ter com uma marca, seus serviços e, principalmente, seus canais digitais, como *sites*, aplicativos e *software*, sendo que essa experiência se inicia desde o primeiro contato com uma marca até o momento do consumo de seus produtos (PEREIRA, 2018).
Existem dois requisitos para que essa experiência seja adequada:

1. deve-se atender às necessidades exatas do usuário,
2. de forma elegante e simples.

A UX não busca somente entregar o que o cliente deseja, mas surpreender o usuário com algo de que nem ele mesmo poderia imaginar que necessitaria.

Além da experiência de usuário, outro importante conceito é o da **interface do utilizador** (do inglês *user interface*, UI), que se refere ao momento específico em que o usuário interage com a interface. Porém, experiência de usuário é um termo mais abrangente, pois envolve a experiência dos usuários marcante na interação com o produto e os serviços da marca, não só em relação a aspectos visuais. O produto deve agradar aos olhos, mas não distrair o usuário do conteúdo e sempre cumprir as funções as quais se propõe (DEVMEDIA, 2013). Também nesse aspecto Steve Jobs foi revolucionário, pois teve a preocupação de reduzir o portfólio de produtos da empresa para facilitar a vida das pessoas no momento da escolha, levando em consideração aspectos para marcar uma boa experiência desde a escolha do produto até a compra (PEREIRA 2018).

Um dos princípios da UX é desenvolver *software* para usuários iniciantes a avançados, todos considerados quando do projeto e da criação do produto

(DEVMEDIA, 2013). Além disso, incluir os princípios da experiência de usuário no desenvolvimento de *software* pode:

- auxiliar a empresa a alavancar ideias;
- ajudar a empresa a descobrir oportunidades de negócio;
- diminuir de custos e retrabalho, evitando que funcionalidades desnecessárias sejam desenvolvidas.

A Figura 2 apresenta as diferentes áreas que compõem a experiência do usuário, como arquitetura de informação, usabilidade, etc.

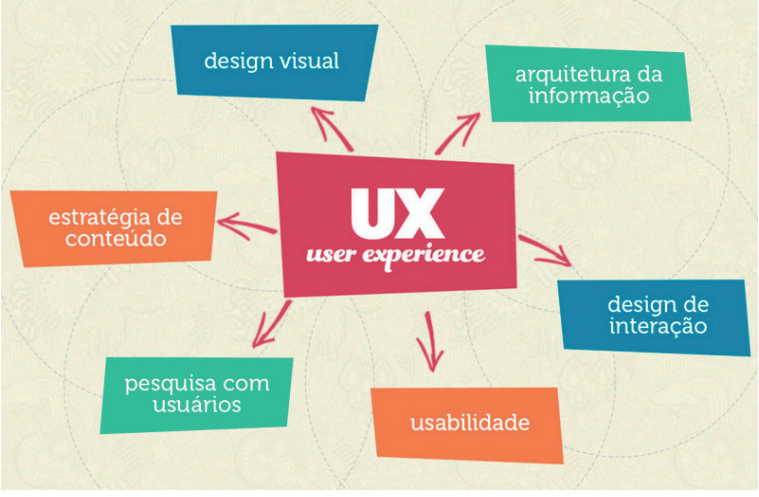

Figura 2. Áreas que envolvem a experiência do usuário.
Fonte: Albuquerque (2015, documento *on-line*).

No desenvolvimento de *software*, o profissional de UX pode aparecer nas três etapas do processo. Veja a seguir.

1. Início do projeto: pode auxiliar na definição dos requisitos, a partir da realização de pesquisas e de entrevistas, e de materiais que possam ajudar a entender melhor o usuário e o problema se resolvido.
2. Desenvolvimento do *software*: nessa etapa, podem ser utilizados aspectos da arquitetura de informação e do *design* de interação para

auxiliar os programadores na definição de fluxos, do comportamento dos componentes, de elementos visuais, etc.

3. Entrega: ao fazer uma análise/avaliação dos padrões adotados a partir de testes de usabilidade e *feedback* dos usuários.

Além disso, existem algumas tendências relacionadas à utilização de UX no desenvolvimento de *software*. Veja a seguir.

- *Design* para aumentar a produtividade: está relacionado a *software* com interfaces simples e agradáveis e com um leiaute objetivo, que não confunda o usuário com várias informações desnecessárias.
- Personalização inteligente: visa permitir que o usuário consiga personalizar o *software* segundo seu perfil, levando em consideração comportamentos e hábitos dos usuários.
- Relação do usuário com o *software*: tem relação com praticar a empatia, ou seja, colocar-se no lugar do usuário para despertar seu interesse.

Nas práticas convencionais de projetos de desenvolvimento de sistemas, a etapa de levantamento de requisitos era conduzida pelos desenvolvedores, analistas de sistemas ou analistas de requisitos, e se buscava trazer os usuários para que relatassem suas necessidades. Essa etapa sofria com a falta de comprometimento dos usuários, porque eles não tinham alocação dedicada para tratar dos requisitos. Assim, um dos grandes problemas na área de desenvolvimento de *software* são requisitos pobremente elicitados e validados por parte dos usuários. Portanto, incluir, de fato, os usuários no processo tende a alavancar o sucesso dessa etapa.

Para a aplicação da UX no desenvolvimento de *software*, pode-se pensar em alguns passos básicos, mesmo que isso deva observar as circunstâncias em que se é aplicada, segundo o perfil do cliente e o tipo de produto. Veja a seguir.

- Estratégia: ter uma visão do projeto, de quais são os objetivos.
- Pesquisa: o foco deve ser o usuário, pois, assim, serão entendidos seus problemas e necessidades e o que se deseja solucionar por meio do *software*.
- Análise: por meio dos dados coletados, são organizadas as hipóteses a serem validadas.
- *Design*: é a criação dos protótipos a partir das hipóteses validadas. Esses protótipos devem ser testados com os usuários a fim de obter os *feedbacks*.

- Produção: depois de validadas as hipóteses e realizado o refinamento do *design* e das funcionalidades, é hora de entregar ao cliente o produto.

Além disso, existem alguns aspectos que devem ser observados na empresa, que são:

- processos colaborativos (todos trabalham juntos, inclusive o cliente);
- comunicação (a comunicação clara e constante é essencial dentro do time e entre times);
- resistência interna (um novo processo pode enfrentar resistência inicial por parte de algumas pessoas, então a sugestão seria iniciar com um projeto pequeno e tentar criar a cultura dentro da empresa pelo *case* de validação).

É necessário ressaltar que basear o desenvolvimento de um *software* no usuário ajuda a definir e tornar mais claro o que o usuário quer e a definir como as pessoas vão interagir com o *software*. Além disso, a UX faz a mediação entre o usuário e o desenvolvedor do projeto, identifica as necessidades, estabelece os requisitos, desenvolve protótipos que possam ser avaliados e acompanha todo o processo de desenvolvimento do projeto (DEVMEDIA, 2013).

Fique atento

Experiência de usuário não é o mesmo que usabilidade. UX trata de como o usuário se sente ao utilizar o sistema, enquanto usabilidade se refere a quão fácil é utilizar sua interface e à eficiência no uso. Assim, a experiência de usuário foca em trazer os principais benefícios da interação entre o usuário e seus serviços, ou seja, em desenvolver soluções intrínsecas ao cotidiano da vida do usuário, de forma a fazer parte do todo, tornando-se imperceptíveis.

3 Técnicas de *design* de interação e experiência de usuário no desenvolvimento ágil

Os métodos ágeis estão evoluindo o tempo todo. As organizações estão reconhecendo que ainda existem questões pendentes, bem como oportunidades para melhorar a forma como o *software* é construído em um ambiente ágil. Os métodos ágeis enfatizam o *software* funcional, mas os usuários ainda esperam um alto nível de usabilidade do *software*. Também é comum que o *design* da experiência do usuário seja uma atividade separada do desenvolvimento da funcionalidade do *software* e frequentemente conduzida por um indivíduo ou uma equipe separada (GREER; HAMON, 2011).

No entanto, existem esforços para realizar a integração entre métodos ágeis e UX. Conforme relatam Rosemberg e Schilling (2011), alguns exemplos podem ser adotados, como:

- sincronizar o trabalho entre o profissional de UX e o time de desenvolvimento (atividades antes, durante e depois de cada *sprint*);
- separar a criação de modelos (usuários, tarefas) do *design* (projeto da solução);
- disseminar a cultura e o conhecimento sobre *design*, UX e IHC no time de desenvolvimento;
- trabalhar colaborativamente (não ser a única fonte de soluções);
- investir em artefatos rápidos/de baixa fidelidade;
- trabalhar com estimativas honestas e claras de esforço e/ou prazo.

Além disso, se as atividades de UX são realizadas em uma *sprint* antes do desenvolvimento, há tempo para testar e criar a melhor solução possível antes que ela seja efetivamente construída, diminuindo gastos e tempo desperdiçados na criação de funcionalidades que não atendam às expectativas. É muito importante que toda a equipe tenha isso em mente.

Saiba mais

Para saber mais sobre a utilização de UX no desenvolvimento de *software*, digite "Como alinhamos UX e *cloud* na construção de um produto escalável?" em seu motor de busca preferido para ter acesso a um interessante estudo de caso de uma empresa farmacêutica.

A utilização de UX no desenvolvimento ágil demanda que (SANCHEZ, 2019):

- a equipe não apenas acredite, mas viva a metodologia;
- exista comunicação, colaboração e confiança dentro da equipe;
- as cerimônias ágeis sejam realizadas de forma consistente e documentadas no tempo certo;
- os membros da equipe sejam transparentes sobre suas preocupações ou bloqueios, e estudem juntos.

Kaley (2019) afirma que os profissionais de UX devem participar de todo o processo. O envolvimento de um profissional de UX em uma equipe ágil significa mais do que apenas trabalhar uma *sprint* à frente para entregar ideias de forma rápida e contínua aos donos de produtos, interessados e desenvolvedores. Significa também ser um colaborador presente e ativo na *sprint* atual, em todas as reuniões e cerimônias. Assim, veja, a seguir, o que o profissional responsável pela UX deve fazer em cada cerimônia *scrum*.

- *Daily scrum*: é a reunião diária, curta, que normalmente não deve passar de 15 minutos e que se concentra na comunicação da equipe, incluindo o profissional de UX, para saber se estão no caminho certo. Normalmente, o profissional de UX está trabalhando uma *sprint* ou mais à frente da equipe de engenharia, então é um desafio manter a equipe alinhada. É por isso que comparecer à reunião diária e comunicar de forma clara e concisa o que o UX está fazendo é importante.
- Refinamento do *backlog*: essa reunião normalmente ocorre no meio da *sprint*, com o objetivo de revisar os itens do *backlog*. Ter profissionais de UX presentes nessa reunião facilita o esboço de técnicas de ideação, então pode ajudar a equipe a resolver problemas em conjunto. A par-

ticipação da UX na preparação do *backlog* pode trazer uma previsão sobre em que a equipe de desenvolvimento trabalhará na próximo *sprint* e fornecer suporte adicional para o *product owner*. A UX deve ajudar a garantir que os proprietários do produto tomem as decisões corretas para os usuários e o produto ao escolher os próximos itens do *backlog*.

- Planejamento da *sprint*: ocorre no primeiro dia da *sprint* ou um dia antes da *sprint* iniciar para planejar as atividades do *backlog*. O envolvimento do profissional de UX no planejamento de *sprint* é importante porque o trabalho dele deve ser contabilizado no *backlog*. Certifique-se de levar em conta os testes do usuário no planejamento da *sprint* para que a equipe saiba o que esperar e tenha a oportunidade de participar das sessões. Caso seja necessário realizar pesquisas maiores ou *workshops*, como *sprints* de *design*, em que é importante que toda a equipe esteja envolvida, deve ser proposta em uma velocidade um pouco mais baixa para essa *sprint*, de modo que os desenvolvedores tenham tempo reservado para participar da pesquisa sem sentir que devem realizar várias tarefas para atingir o objetivo da *sprint*.
- Revisão da *sprint*: geralmente ocorre no penúltimo dia da *sprint*. Em algumas vezes, as demandas da UX podem deixar a demonstração da *sprint* sobrecarregada devido a várias solicitações de novos recursos ou pelos *feedbacks* recebidos.
- Retrospectiva: geralmente ocorre no último dia da *sprint* e tem como objetivo revisar, junto com a equipe, como ocorreram as coisas no decorrer do processo. É um ótimo momento para o profissional encarregado pela UX falar sobre as mudanças no processo. Em ambientes focados no desenvolvedor, pode ser difícil defender mudanças no processo que vão melhorar a capacidade de entregar um bom trabalho de UX, então deve ser considerado posicionar as ideias do UX como experimentos a serem testados durante a próxima *sprint*.

Para Rogers, Sharp e Preece (2013), a utilização do *design* de interação no desenvolvimento ágil facilita alguns processos, como:

- o processo de prototipação, pois, por meio dos protótipos, os usuários já podem dar *feedbacks* que permitem as melhorias e os ajustes necessários;
- a participação ativa dos usuários que utilizarão o sistema, ou seja, que têm a perspectiva da pessoa que vai utilizar o sistema todos os dias, não somente dos interessados no projeto;

- os diferentes graus de envolvimento do usuário, em que se busca envolver diversos usuários, a exemplo de empresas que têm funcionários em tempo parcial e integral;
- o envolvimento dos usuários após o lançamento do produto, pois, por meio desse processo, é possível obter *feedbacks* dos usuários em relação ao produto real.

Por fim, existem alguns tipos de apoio para o *design*, como os listados a seguir (ROGERS; SHARP; PREECE, 2013).

- Utilização de padrões de projetos para o *design* de interação: utilizar padrões para auxiliar na criação de *software*, já que é uma solução para um problema em determinado contexto. Os usuários do padrão podem entender em que circunstâncias essa solução foi aplicada.
- Sistemas e componentes de código aberto: existem vários componentes de *software* e *hardware* sob licença de código aberto para que se possa modificá-los conforme a necessidade. Pode-se tentar encontrar um projeto nesse formato que se adéque ao problema a ser resolvido e utilizá-lo, fazendo apenas as modificações e adaptações necessárias.
- Ferramentas e ambientes: existem várias ferramentas para apoiar o pensamento criativo, como esboços de telas, simulações, etc. Pelo menos uma ferramenta será utilizada para o processo de *design*, a exemplo de *sites* para criação de interface *mobile*, *web design*, desenvolvimento de robôs e assim por diante.

O envolvimento dos usuários no processo de *design* de interação faz os *software* evoluírem de algumas ideias iniciais para *designs* conceituais e protótipos. Com isso, os protótipos podem fazer parte dos diversos ciclos existentes no desenvolvimento por meio da utilização de um *framework* ágil. Porém, deve-se observar quando o protótipo já está se parecendo com o produto para que seja encerrada a prototipação e se passe para o desenvolvimento.

Tanto na UX quanto no *design* de interação, o foco principal está no usuário. Atender às suas expectativas e reais necessidades é o principal benefício obtido por meio da aplicação desses conceitos. É importante entender os objetivos que norteiam um bom projeto, identificando seu valor, a praticidade no uso do produto e uma boa experiência para o usuário.

Saiba mais

Você conhece ScrumUX? **ScrumUX** é uma abordagem para integrar *design* de interação e metodologias ágeis, mais especificamente o processo *scrum*. Para saber mais a respeito, digite "ScrumUX: uma abordagem para integrar *design* de interação do usuário ao processo *scrum*" em seu motor de busca preferido para ter acesso a uma tese que trata do assunto.

Referências

DEVMEDIA. User Experience UX: Ajudando no desenvolvimento de software. Devmedia, 2013. Disponível em: https://www.devmedia.com.br/user-experience-ux-ajudando-no-desenvolvimento-de-software/28872. Acesso em: 01 ago. 2022.

ALBUQUERQUE, P. *Em que consiste uma boa experiência de usuário*. 2015. Disponível em: http://catarinasdesign.com.br/em-que-consiste-uma-boa-experiencia-do-usuario/. Acesso em: 21 set. 2020.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *ABNT NBR ISO 9241-110:2012*: ergono-mia da interação humano-sistema, parte 110 – princípios de diálogo. Rio de Janeiro: ABNT, 2012.

BERNARDINO, C. B. *Design interativo em processos ágeis de desenvolvimento de software*. 2005. 83 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) — Centro de Informática. Universidade Federal de Pernambuco, Recife, 2005.

CYBIS, W.; BETIOL, A. H.; FAUST, R. *Ergonomia e usabilidade*: conhecimentos, métodos e aplicações. São Paulo: Novatec, 2017.

FERREIRA, J.; NOBLE, J.; BIDDLE, R. Up-front interaction design in agile development. *In*: INTERNATIONAL CONFERENCE ON EXTREME PROGRAMMING AND AGILE PROCESSES IN SOFTWARE ENGINEERING, 8., 2007, Como. *Anais* [...]. Heidelberg: Springer, 2007. p. 9–16.

GREER, D.; HAMON, Y. Agile software development. *Software: Practice and Experience*, v. 41, nº. 9, p. 943–944, 2011.

KALEY, A. *UX Responsibilities in Scrum Ceremonies*. 2019. Disponível em: https://www.nngroup.com/articles/ux-scrum/. Acesso em: 21 set. 2020.

PEREIRA, R. *User experience design*: como criar produtos digitais com foco nas pessoas. São Paulo: Casa do Código, 2018.

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

ROSEMBERG, C.; SCHILLING, A. Integrando IHC e métodos ágeis. *In*: LATIN AMERICAN CONFERENCE ON HUMAN-COMPUTER INTERACTION, 5., 2011, Porto de Galinhas. *Anais* [...]. Porto Alegre: Brazilian Computer Society, 2011. p. 36–38.

SANCHEZ, M. *UX e Métodos Ágeis*. 2019. Disponível em: https://uxpmbrasil.com.br/ux-e-agile-de-m%C3%A3os-dadas-9001726e3b14. Acesso em: 21 set. 2020.

VILACA, T. *Um panorama sobre o design da Apple*: o design que mudou os padrões de consumo. 2017. Disponível em: https://www.alura.com.br/artigos/um-panorama-sobre-o-design-da-apple-o-design-que-mudou-os-padroes-de-consumo. Acesso em: 21 set. 2020.

Leituras recomendadas

ANDROID DEVELOPERS. *Projetar para Android*. c2020. Disponível em: https://developer.android.com/design. Acesso em: 21 set. 2020.

NIELSEN, J. *Usability engineering*. Mountain View: Morgan Kaufmann, 1994.

SCAPIN, D. L.; BASTIEN, J. M. C. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. *Behaviour & information technology*, v. 16, nº. 4–5, p. 220–231, jul. 1997.

SHNEIDERMAN, B.; PLAISANT, C. *Designing the user interface*: strategies for effective human-computer interaction. New York: Pearson, 2010.

Fique atento

Os *links* para *sites* da *web* fornecidos neste capítulo foram todos testados, e seu funcionamento foi comprovado no momento da publicação do material. No entanto, a rede é extremamente dinâmica; suas páginas estão constantemente mudando de local e conteúdo. Assim, os editores declaram não ter qualquer responsabilidade sobre qualidade, precisão ou integralidade das informações referidas em tais *links*.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS