

# Uma análise exploratória de ferramentas de prototipação de *software*

Rafael Felipe dos Santos Machado  
rafael.machado@sou.unifal-mg.edu.br  
Universidade Federal de Alfenas  
Alfenas, Minas Gerais, Brasil

Mariane Moreira de Souza  
mariane.souza@unifal-mg.edu.br  
Universidade Federal de Alfenas  
Alfenas, Minas Gerais, Brasil

## RESUMO

A prototipação é uma etapa essencial no contexto de UX/UI (*User eXperience/User Interface*), uma vez que facilita o diálogo entre os interessados e coleta requisitos importantes do usuário para o processo de desenvolvimento de *software*. Apesar de existirem diversas ferramentas digitais que facilitam a criação de protótipos a nível profissional, estudos exploratórios sobre as principais características das ferramentas presentes no mercado atualmente são bastante escassos. Este trabalho pesquisa e analisa ferramentas para o processo de prototipação, com base em parâmetros previamente definidos, com o objetivo de facilitar a escolha futura de ferramentas em um ambiente real. Por meio da análise das ferramentas, foram encontrados dados concretos sobre suas vantagens e desvantagens, que podem guiar o responsável pela etapa de prototipação a fazer melhores escolhas.

## CCS CONCEPTS

• **Human-centered computing** → **Systems and tools for interaction design**.

## KEYWORDS

Prototipação, UX/UI, *design*, desenvolvimento de *software*

### ACM Reference Format:

Rafael Felipe dos Santos Machado and Mariane Moreira de Souza. 2021. Uma análise exploratória de ferramentas de prototipação de *software*. In *IHC '21: 20th Brazilian Symposium on Human Factors in Computing Systems, October 18–22, 2021, Online, Brazil*. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/1122445.1122456>

## 1 INTRODUÇÃO

Muito se fala sobre experiência do usuário (UX) e interface do usuário (UI) no desenvolvimento de *software* e sobre como a prototipação ajuda na construção de ambientes virtuais que proporcionam cada vez mais uma interface intuitiva e uma experiência recompensadora [9]. Além disso, a metodologia de desenvolvimento de *software* orientada à prototipação é complementar aos métodos baseados em ciclo de vida, devido à sua natureza de especificar onde e como um

ciclo é possível e necessário, segundo a avaliação dos protótipos a cada momento [18].

Atualmente, existem várias ferramentas no mercado que apoiam a construção de protótipos [20] [15] [13] [11] [14] [12] [7] [1]. Apesar disso, poucos trabalhos discutem as características, vantagens e desvantagens de ferramentas de prototipação. Quando existentes, tais estudos contemplam poucas ferramentas, consideradas defasadas devido à época do estudo e ao rápido crescimento tecnológico [4] [8].

Este trabalho pesquisa e analisa as principais ferramentas de prototipação disponíveis no mercado atualmente, classificando-as de acordo com seus pontos positivos e negativos, segundo critérios pré-definidos. A partir dos resultados dessa análise exploratória, é possível obter uma base de comparação, que funciona como um guia para a escolha da(s) melhor(es) ferramenta(s) em um caso real. Com o objetivo de verificar como os resultados dessa análise poderiam ser aplicados na prática, uma pesquisa-ação foi realizada, considerando a escolha das melhores ferramentas para a etapa de prototipação de um aplicativo hipotético para adoção de animais de estimação.

## 2 PROTOTIPAGEM DE SOFTWARE

A prototipação de *software* está diretamente relacionada com o conceito de experiência do usuário (UX), afetando positivamente o desenvolvimento de interfaces de usuário (UI) [17]. Segundo Garrett [9], experiência do usuário é "a experiência que o produto cria para as pessoas que o usam no mundo real". Reconhecendo a importância dessa experiência, cada vez mais empresas têm investido na construção de protótipos, com o objetivo de capturar informações para fornecer uma boa experiência do usuário, considerando ser essa uma vantagem essencial para o negócio de todo tipo de produto e serviço [9].

A prototipação é uma importante etapa complementar do processo de desenvolvimento de *software* [18]. Segundo Budde e Stage [3], um protótipo é um modelo funcional de uma aplicação; executa certos aspectos a serem implementados no produto final e fornece uma base concreta para discussões entre desenvolvedores, clientes e outros interessados. Além disso, os protótipos apoiam a criatividade, ajudando o desenvolvedor a capturar e gerar ideias, além de auxiliar na descoberta de informações relevantes sobre os usuários e suas práticas de trabalho [4].

Böhmer, et al. [6] abordam as diferentes classificações de protótipos, incluindo, dentre outros aspectos, os tipos e objetivos do protótipo. Möller (apud 2011) [16] diferencia quatro tipos de protótipos: modelos de conceito, protótipos geométricos, protótipos funcionais e protótipos técnicos. Hallgrimsson (apud 2012) [10]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*IHC '21, October 18–22, 2021, Online, Brazil*

© 2021 Association for Computing Machinery.  
ACM ISBN 978-1-4503-XXXX-X/18/06... \$15.00  
<https://doi.org/10.1145/1122445.1122456>

classifica os objetivos do protótipo em quatro categorias principais: exploração; comunicação; usabilidade e *design*; e verificação. Thomke (apud 2007) [19] diferencia os protótipos em de baixa e de alta fidelidade. Protótipos de baixa fidelidade são baratos e podem ser produzidos rapidamente para um *feedback* já na fase inicial do processo de desenvolvimento (apud THOMKE, 2007) [19]. Protótipos de alta fidelidade são cada vez mais importantes para entender o quão perto de uma solução está o esforço do projeto e para minimizar o risco de falhas (apud THOMKE, 2007) [19]. Protótipos de média fidelidade são, por sua vez, um intermediário entre os dois.

Neste trabalho, temos como objeto de estudo ferramentas digitais que possibilitam a prototipação funcional de baixa, média e alta fidelidade para o *design* de interfaces de aplicações *web* e *mobile*. As ferramentas escolhidas foram: ProtoPie [20], Justinmind [15], InVision [13], Axure [11], Webflow [14], Figma [12], Sketch [7] e Adobe XD [1]. Mais detalhes sobre a escolha e avaliação das ferramentas são apresentados na Seção 3.

### 3 AVALIAÇÃO DAS FERRAMENTAS

A falta de informações concretas sobre ferramentas de prototipação pode dificultar, especialmente para empresas inexperientes na área, a adoção de uma ou outra ferramenta. Nesta seção mostramos uma alternativa para apoiar essa escolha, por meio da avaliação de ferramentas de prototipação, possibilitando que a empresa poupe tempo e esforço durante a etapa de escolha das ferramentas e os invista na criação do protótipo e no desenvolvimento do produto propriamente dito.

#### 3.1 Escolha dos parâmetros

Para atingir um resultado concreto na avaliação das ferramentas, definimos as características que julgamos mais importantes quando pensamos em uma ferramenta de prototipação, sendo elas:

- (1) Nível de fidelidade (Baixa/Média/Alta): o nível de fidelidade representa o quão próximo do produto final o protótipo está. É importante saber se a ferramenta fornece apoio, em especial, para o desenvolvimento de protótipos de alta fidelidade, que são aqueles que pretendem representar com maior fidelidade o comportamento real do futuro produto a ser desenvolvido;
- (2) Interatividade (Estático/Interativo): é relevante saber se a ferramenta fornece apoio à modelagem de interatividade, visto que o produto final representa uma experiência de usuário, com diversas interações;
- (3) Colaboratividade (Possui/Não Possui): característica que considera a predominância de ambientes colaborativos de desenvolvimento de *software* [5], em que os desenvolvedores trabalham paralelamente, de forma a colaborar para o desenvolvimento final do produto. Caso a ferramenta possua formas de colaboratividade, na tabela também é informado, entre parenteses, se esta está presente também em sua versão gratuita e/ou no teste grátis, ou apenas na versão paga;
- (4) Portabilidade (Windows/Linux/macOS/Android/iOS/Web): devido à variabilidade de sistemas operacionais existentes atualmente, é necessário saber em quais deles é possível executar a ferramenta;

- (5) Integração com outras ferramentas (Sim/Não): como as ferramentas possuem vantagens e desvantagens entre si, muitas vezes é interessante escolher mais de uma ferramenta para prototipar. Para isso, é fundamental saber se a ferramenta possui integração com outras ferramentas de prototipação e, se sim, quais;
- (6) Facilidade de aprendizado (Fácil/Média/Difícil): é importante entender se a ferramenta possui uma interface de fácil aprendizado, diminuindo o esforço necessário na prototipação;
- (7) Rapidez no desenvolvimento (Devagar/Média/Rápida): também é útil observar se a ferramenta oferece formas simples e rápidas de se fazer o que precisa ser feito, diminuindo o tempo necessário para prototipar;
- (8) Custo (Valores): essa é uma característica primordial a ser considerada, visto que muitas ferramentas no mercado são pagas e nem sempre a empresa de desenvolvimento pode arcar com esses custos;
- (9) Suporte a *wireframes* (desenho básico usado para demonstrar a arquitetura do modelo) (Sim/Não): considerando a escolha pelo uso de ferramentas de apoio para protótipos de baixa fidelidade, é essencial que o suporte a *wireframes* seja considerado.
- (10) Exportação para código (Sim/Não): característica que considera a busca de ferramentas de prototipação que ofereçam a possibilidade de converter o protótipo criado em código executável, a fim de economizar tempo e esforço na implementação do *software*;
- (11) Possui teste grátis (Sim/Não/Totalmente Grátis): como muitas ferramentas no mercado são pagas, é pertinente considerar se estas oferecem algum tempo de teste inicial, para que o responsável pela prototipação possa analisar na prática se é viável ou não pagar pelo serviço oferecido. Caso a ferramenta não seja paga ou possua uma versão gratuita (sem prazo de término), esta é avaliada como "Versão Gratuita".

#### 3.2 Avaliação das ferramentas

Durante a etapa de avaliação, as ferramentas foram analisadas em relação a cada parâmetro mencionado na Seção 3.1. A Tabela 1 mostra os resultados obtidos após essa avaliação.

É importante frisar que os parâmetros "Facilidade de aprendizado" (6) e "Rapidez no desenvolvimento" (7) são de natureza subjetiva e foram, portanto, analisados e documentados de acordo com a experiência obtida pelo autor deste trabalho. Por este motivo, sugere-se como estudos futuros a quantificação destes parâmetros.

Por último, consideramos os parâmetros "Nível de fidelidade" (1) e "Interatividade" (2), para a ferramenta InVision, que oferece dois ambientes distintos para *wireframes* e protótipos. Como o nível de fidelidade e a interatividade são muito diferentes entre os ambientes de prototipação e de *wireframes*, consideramos que seria de maior valor disponibilizar ambas as avaliações na Tabela 1.

Na próxima seção, apresentamos um exemplo de aplicação dos resultados da avaliação na escolha de ferramentas de prototipação para o desenvolvimento de um aplicativo hipotético.

Tabela 1: Resultados obtidos na análise das ferramentas de prototipação

	ProtoPie [20]	Justinmind [15]	InVision [13]	Axure [11]	Webflow [14]	Figma [12]	Sketch [7]	Adobe XD [1]
<b>Nível de fidelidade</b>	Alta	Alta	Baixa ( <i>wireframes</i> ) / Alta (protótipos)	Alta	Alta	Média	Alta	Alta
<b>Interatividade</b>	Interativo	Interativo	Estático ( <i>wireframes</i> ) / Interativo (protótipos)	Interativo	Interativo	Interativo	Interativo	Interativo
<b>Colaboratividade</b>	Possui (apenas pago)	Possui (também no teste grátis)	Possui (também no gratuito)	Possui (também no teste grátis)	Possui (apenas pago)	Possui (também no gratuito)	Possui (também no teste grátis)	Possui (também no teste grátis)
<b>Portabilidade</b>	Windows, macOS	Windows, macOS	Web, Windows, macOS	Windows, macOS	Web	Web, Windows, macOS	macOS	Windows, macOS
<b>Integração com outras ferramentas</b>	Sim (Sketch, Adobe XD, Figma)	Sim (Sketch, Photoshop, Illustrator, Adobe XD, Atlassian Jira, Microsoft Team Foundation Server, user testing tools e PhoneGap)	Sim (Sketch e Photoshop)	Sim (Sketch e Axure Cloud)	Não	Sim (Sketch)	Não	Sim (todas da Adobe e Sketch)
<b>Facilidade de aprendizado</b>	Fácil	Fácil	Fácil ( <i>wireframes</i> ) / Difícil (protótipos)	Difícil	Fácil	Fácil	Fácil	Fácil
<b>Rapidez no desenvolvimento</b>	Rápida	Rápida	Média	Devagar	Rápida	Rápida	Rápida	Rápida
<b>Custo</b>	US\$13/mês ( <i>Individual</i> ), US\$50/mês ( <i>Team</i> )	US\$19/mês ( <i>Professional</i> ), US\$39/mês ( <i>Enterprise</i> )	Grátis, US\$9.95/mês ( <i>Pro</i> )	US\$25/mês ( <i>Pro</i> ), US\$42/mês ( <i>Team</i> )	Grátis, US\$16/mês ( <i>Lite</i> ), US\$35/mês ( <i>Pro</i> )	Grátis, US\$12/mês ( <i>Professional</i> ), US\$45/mês ( <i>Organization</i> )	US\$99 Life-time ( <i>Individual</i> ), US\$9/mês para cada contribuinte ( <i>Team</i> )	R\$43/mês ( <i>Individual</i> ), R\$224/mês ( <i>Completo</i> )
<b>Suporte a <i>wireframes</i></b>	Não	Sim	Sim	Não	Não	Sim	Sim	Sim
<b>Exportação para código</b>	Não	Sim	Não	Sim	Pago	Não	Sim	Sim (com <i>plugin</i> de terceiros)
<b>Possui teste grátis</b>	Sim	Sim	Versão Gratuita	Sim	Versão Gratuita	Versão Gratuita	Sim	Sim

## 4 ESCOLHA DE FERRAMENTAS E PROTOTIPAÇÃO DO APLICATIVO

Com o objetivo de verificar na prática os resultados da análise exploratória das ferramentas, aplicamos estes resultados na etapa de prototipação de um aplicativo de adoção de animais de estimação em situação de rua, nomeado Adote, devido à demanda encontrada em Alfenas, Minas Gerais.

Após o levantamento dos requisitos iniciais do produto (feito pela coautora deste artigo, Mariane Moreira de Souza, e que pode ser acessado no [link](https://tcc.rafaelmachado.dev) <https://tcc.rafaelmachado.dev>), o próximo passo foi a escolha das ferramentas, considerando as etapas de um processo de prototipação (na ordem da mais baixa à mais alta fidelidade) [2]:

- (1) Criar *wireframes* (etapa relativa à prototipação de baixa fidelidade): antes de construir um protótipo de alta fidelidade, é importante pensar na estrutura básica da interface, definir as telas e os componentes principais que devem estar presentes no produto final, analisar as possíveis disposições dos componentes e documentar o fluxo de telas. A Figura 1 ilustra uma parte dos *wireframes* criados, utilizando a ferramenta *web* InVision. A ferramenta foi escolhida por ser de uso gratuito, pela portabilidade *web* e pelo bom suporte ao desenho de *wireframes*, como levantado na Tabela 1. O resultado de todos os *wireframes* pode ser acessado no [link](https://tcc.rafaelmachado.dev) <https://tcc.rafaelmachado.dev>.

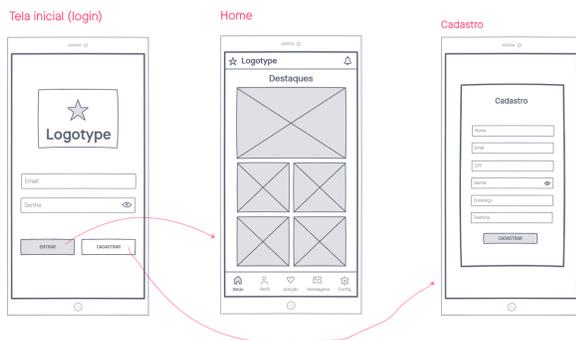


Figura 1: Exemplo de *wireframes* (tela inicial, tela de destaques e tela de cadastro) criados com a ferramenta Invision [13].

- (2) Definir paleta de cores: antes de prosseguir para as etapas de fidelidade média e alta, é útil definir uma paleta harmônica e utilizá-la durante o processo para padronizar a identidade visual do produto. A Figura 2 mostra as cores escolhidas para o protótipo do nosso aplicativo hipotético.
- (3) Criar *mockup*: esta etapa é relativa à prototipação de média fidelidade, que se dá por meio da criação do *mockup*, representando um modelo do produto final. Nesta etapa, desenhamos as telas estabelecidas na etapa 1, utilizando as cores definidas na etapa 2. Algumas interações, como troca de telas, foram modeladas, iniciando o projeto de interações do aplicativo. Para essa etapa escolhemos a ferramenta Figma devido à gratuidade, portabilidade, facilidade de aprendizado, rapidez no desenvolvimento e ao suporte a desenhos



Figura 2: Paleta de cores escolhida para prototipar o aplicativo. Fonte: Adaptado de *Working With Dog* de J. Nichole Smith (<https://workingwithdog.com/color/>).

de média fidelidade. A Figura 3 mostra um exemplo das telas criadas para o *mockup* do aplicativo. O resultado final com todas as telas do *mockup* pode ser acessado no [link](https://tcc.rafaelmachado.dev) <https://tcc.rafaelmachado.dev>.

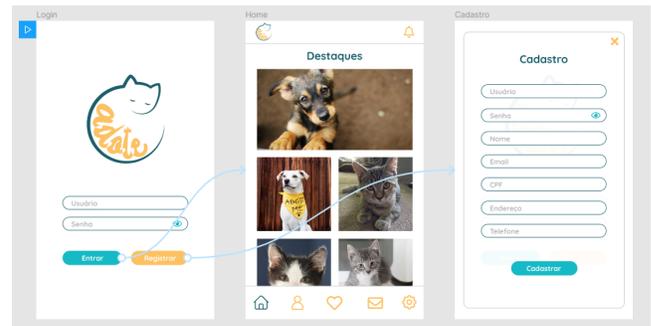
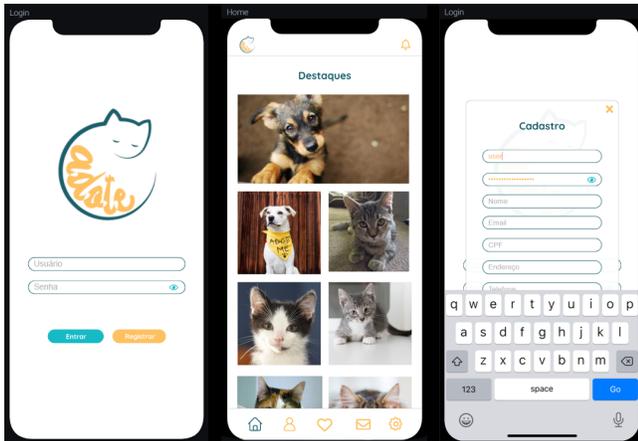


Figura 3: Exemplo de telas (tela inicial, tela de destaques e tela de cadastro) do *mockup* do aplicativo Adote, criado com a ferramenta Figma [12].

- (4) Criar protótipo (etapa relativa à prototipação de alta fidelidade): após desenhar as telas com cores e interações básicas, foram criadas as interações mais complexas, como entradas de texto interativas, efeitos, movimentos etc. Isso foi feito para tornar o protótipo o mais próximo possível do produto final e possibilitar futuros testes da UX/UI antes de implementar qualquer código. Para esta etapa, escolhemos a ferramenta ProtoPie, devido ao seu alto nível de fidelidade, rapidez no desenvolvimento, facilidade de aprendizado, boa portabilidade, disponibilidade de teste grátis e integração com a ferramenta Figma, que utilizamos na criação do *mockup*. A Figura 4 ilustra o mesmo conjunto de telas da etapa anterior, mas agora na fase de protótipo. O resultado final com todas as telas do protótipo pode ser acessado no [link](https://tcc.rafaelmachado.dev) <https://tcc.rafaelmachado.dev>.

## 5 RESULTADOS E DISCUSSÕES

Os dados da Tabela 1 mostram a inexistência de uma ferramenta que só tenha vantagens considerando todos os parâmetros, mas a existência de ferramenta(s) que proporcione(m) vantagens que melhor atende(m) a demanda de cada empresa. Por exemplo, a



**Figura 4: Exemplo de telas (tela inicial, tela de destaques e tela de cadastro) do protótipo do aplicativo Adote, criado com a ferramenta ProtoPie [20].**

ferramenta Adobe XD apresenta um bom resultado com relação à maioria dos parâmetros, mas possui um custo elevado em sua versão completa, considerando empresas que tenham outras prioridades de investimento. Outro exemplo considera as ferramentas Protopie e Justinmind, que possuem avaliações semelhantes, diferindo em alguns parâmetros, como colaboratividade e custos, o que poderia levar à escolha da ferramenta Justinmind como melhor opção. Com relação à ferramenta Axure, que apresenta aprendizado difícil com relação às demais, não seria indicada para empresas em que os profissionais ainda não possuem experiência em prototipação, dentre outras análises. De maneira geral, podemos concluir também que pode ser mais interessante que a empresa escolha uma ferramenta que possibilite a geração de ideias mais rapidamente, trazendo um *feedback* do cliente para o problema de *design* em questão.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho tem como contribuição a análise exploratória de ferramentas de prototipação, com o objetivo de auxiliar profissionais da área de desenvolvimento de *software* na escolha de ferramentas, na prática. A partir da aplicação dos resultados da análise, pudemos observar na prática a facilitação no processo de escolha das ferramentas para a prototipação do aplicativo hipotético Adote.

Como trabalhos futuros, sugere-se a utilização dos resultados da avaliação na etapa de prototipação em uma empresa de desenvolvimento de *software*. Os resultados também serão utilizados na escolha de ferramentas a serem utilizadas para o ensino de aspectos de UX, em uma disciplina de IHC da graduação do curso de Bacharelado em Ciência da Computação pela Universidade Federal de Alfenas (Unifal-MG). Além disso, também é sugerido um trabalho de quantificar os parâmetros subjetivos "Facilidade de aprendizado"(6) e "Rapidez no desenvolvimento"(7).

Informações mais completas sobre o trabalho podem ser encontradas na *link* <https://tcc.rafaelmachado.dev>. A Figura 5 mostra um *QR code* que leva a este *link* a fim de facilitar o acesso em caso de leitura via artigo impresso.



**Figura 5: QR code que leva ao link <https://tcc.rafaelmachado.dev> com informações mais específicas e outras mídias do aplicativo Adote.**

## REFERÊNCIAS

- [1] Adobe. 2021. Adobe XD. <https://www.adobe.com/products/xd.html>
- [2] Jonathan Arnowitz, Michael Arent, and Nevin Berger. 2010. *Effective prototyping for software makers*. Elsevier.
- [3] Richard L Baskerville and Jan Stage. 1996. Controlling prototype development through risk analysis. *Mis Quarterly* (1996), 481–504.
- [4] Michel Beaudouin-Lafon and Wendy E Mackay. 2007. Prototyping tools and techniques. In *The human-computer interaction handbook*. CRC Press, 1043–1066.
- [5] Susanne Bødker, Pelle Ehn, Joergen Knudsen, Morten Kyng, and Kim Madsen. 1988. Computer support for cooperative design. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*. 377–394.
- [6] Annette Isabel Böhmer, Rafael Hostettler, Christoph Richter, Udo Lindemann, Jörg Conrad, Alois Knoll, et al. 2017. Towards Agile Product Development-The Role of Prototyping. In *DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, Canada, 21-25.08. 2017*. 001–010.
- [7] Sketch B.V. 2021. Sketch. <https://www.sketch.com/>
- [8] Alan M Davis. 1995. Software prototyping. In *Advances in computers*. Vol. 40. Elsevier, 39–63.
- [9] Jesse James Garrett. 2010. *The elements of user experience: user-centered design for the web and beyond*. Pearson Education.
- [10] Bjarki Hallgrímsson. 2012. *Prototyping and modelmaking for product design*. Hachette UK.
- [11] Axure Software Solutions Inc. 2002-2021. Axure. <https://www.axure.com/>
- [12] Figma Inc. 2021. Figma. <https://www.figma.com/>
- [13] InVisionApp Inc. 2021. InVision. <https://www.invisionapp.com/>
- [14] Webflow Inc. 2021. Webflow. <https://webflow.com/>
- [15] Justinmind. 2021. Justinmind. <https://www.justinmind.com/>
- [16] E Möller. 2011. Handbuch Konstruktionswerkstoffe-Auswahl, Anwendung.
- [17] Chris Northwood. 2018. User Experience. In *The Full Stack Developer*. Springer, 47–66.
- [18] Gustav Pomberger, Walter R. Bischofberger, Dieter Kolb, Wolfgang Pree, and Holger Schlemm. 1991. Prototyping-Oriented Software Development - Concepts and Tools. *Structured Programming* 12, 1 (1991), 43–60.
- [19] Stefan Thomke. 2007. Learning by experimentation: Prototyping and testing. In *Handbook of new product development management*. Routledge, 417–436.
- [20] Studio XID. 2021. ProtoPie. <https://www.protopie.io/>