

Pablo Perea, Pau Giner

UX Design

Projektowanie
aplikacji dla
urządzeń mobilnych

Tytuł oryginału: UX Design for Mobile

Tłumaczenie: Piotr Rajca

ISBN: 978-83-283-4699-4

Copyright © Packt Publishing 2017. First published in the English language under the title 'UX Design for Mobile (9781787283428)'

Polish edition copyright © 2019 by Helion SA
All rights reserved.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Helion SA dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Helion SA nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Helion SA

ul. Kościuszki 1c, 44-100 Gliwice

tel. 32 231 22 19, 32 230 98 63

e-mail: helion@helion.pl

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/uxdesi>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

O autorach	11
O recenzencie	12
Wstęp	13
Rozdział 1. Zasady projektowe i planowanie projektu ogólnego	17
Zmiana perspektywy i skoncentrowanie uwagi na użytkowniku	19
Przekonywanie organizacji, by przyjęły podejście skoncentrowane na użytkowniku	20
Zasady projektowe	22
Potrzeby związane z naszymi ludzkimi cechami	23
Potrzeby związane z ogólnymi oczekiwaniami	27
Potrzeby wynikające z konkretnego kontekstu zastosowania	34
Planowanie projektu ogólnego	38
W praktyce	39
Projektowanie wcale nie polega na tym, by produkty wyglądały ładnie	39
Użytkownicy nie są w stanie nam powiedzieć, czego potrzebują	40
To nie Ty (ani Twój zespół) jesteś użytkownikiem	40
Doświadczenia użytkownika to nie lista możliwości	40
Twoim celem nie jest uszczęśliwienie klienta lub szefa	41
Podsumowanie	41
Rozdział 2. Badanie — uczenie się od użytkowników	43
Jak to zrobić?	44
Poznanie użytkowników	44
Model umysłowy — zrozumieć, jak myślą użytkownicy	44
Czym jest model pojęciowy?	46

Metody badawcze	47
Obserwacja i wywiady	47
Diagramy pokrewieństwa	48
Sortowanie kart	52
Test drzewa	55
Ankiety	58
Dokumenty badawcze	62
Dokument osoby	63
Scenariusze	68
Podróż użytkownika	69
Analiza konkurencyjnych produktów	71
Użytkownicy docelowi, użytkownicy obsługiwani oraz użytkownicy nieobsługiwani	72
W praktyce	73
Rekrutacja uczestników badań	73
Zachowanie obiektywizmu	74
Wypatrywanie znaków	74
Podsumowanie	75
Rozdział 3. Badanie pomysłów — szkicowanie	77
Z odpowiednim podejściem do szkicowania	78
Podstawy szkicowania	80
Narzędzia do szkicowania	81
Szybkie efekty	83
Koncentracja na pracy	83
Określanie limitów czasowych z wykorzystaniem metody szalonych ósemek	84
Organizacja badań	86
Rozkład problemu	86
Identyfikacja konfliktów	87
Wyrażanie swoich pomysłów	89
Tworzenie scenarysów	89
Organizowanie sesji analizy krytycznej	90
W praktyce	92
Lepsze jest wrogiem dobrego	92
Nie chodzi o inspirację	93
Nawiązywanie w rozmowie do celów projektu	93
Uwidacznianie kompromisów — nie wszystko można zaakcentować	94
Podsumowanie	95
Rozdział 4. Wzorce mobilne — najlepsze praktyki dla aplikacji internetowych na Androida i iOS-a	97
Sposoby tworzenia aplikacji mobilnych — responsywność, adaptacja i elastyczność	99
Responsywność działania	99
Strony płynne	99
Strony adaptacyjne	100
Rozwiązania hybrydowe	101
Frameworki do tworzenia interfejsu użytkownika	101
Projektowanie w oparciu o kolumny	102

Wzorce projektowe dla aplikacji mobilnych	103
Wzorce projektowe związane z interfejsem użytkownika aplikacji mobilnych	104
Wzorce projektowe zachowań na urządzeniach mobilnych	122
Różnice pomiędzy Androidem i iOS-em	125
Przyciski fizyczne	126
Gęstość i cechy ekranów	126
Elementy interfejsu użytkownika	127
W praktyce	129
Badanie różnych rozwiązań	129
Podsumowanie	130
Rozdział 5. Uszczegóławianie rozwiązania — szkielety i makiety	131
Wyjaśnienie, czym są szkielety	132
Szkielety jako specyfikacja możliwości funkcjonalnych	132
Układ	134
Gesty	135
Przejścia	136
Makiety	141
Testowanie makiet z wykorzystaniem rzeczywistych treści	142
Specyfikacje i zasoby dla różnych ekranów	143
W praktyce	153
Nauka kodowania	154
Testowanie w rzeczywistych środowiskach	154
Znajomość aktualnych trendów	154
Podsumowanie	155
Rozdział 6. Prototypowanie — ożywianie pomysłów	157
Z odpowiednim podejściem do prototypowania	158
Dobór sposobu prototypowania najlepiej spełniającego nasze potrzeby	159
Wybór typu prototypu	161
Dobór narzędzi do tworzenia prototypów	165
Planowanie prototypu	169
Określanie, co ma zostać uwzględnione w prototypie	169
Wybór sposobu tworzenia prototypu	170
W praktyce	171
Prototyp to nie wczesna wersja produktu	171
Prototyp jest wart tysiąca spotkań	172
Podsumowanie	172
Rozdział 7. Prototypowanie z wykorzystaniem ruchu — stosowanie Tumult Hype	173
Znaczenie ruchu w prototypach	175
Podstawy programu Hype	176
Lista scen	177
Edytor sceny	177
Inspektor właściwości	177
Oś czasu i warstwy	178
Edytor kodu	180
Importowanie i eksportowanie	180

Prototypowanie nawigacji	181
Interakcje operujące na warstwach	188
Obsługa przewijania i nawigacji	195
Obsługa gestów	201
Prototypowanie złożonych zachowań	205
Łączenie animacji	205
Obsługa wprowadzania danych przez użytkownika	211
W praktyce	217
Zapewnienie organizacji warstw	217
Częste przeglądanie efektów	218
Podsumowanie	218
Rozdział 8. Prototypowanie z użyciem kodu — stosowanie programu Framer Studio	219
Framer.js	220
Framer Studio	220
Wprowadzenie do języka CoffeeScript	221
Zacznijmy programować w języku CoffeeScript	222
Zmienne i operatory	222
Typy	224
Funkcje	224
Pętle i konstrukcje warunkowe	225
Obiekty i metody	227
Prezentacja Framer.js	228
Warstwy	229
Animacje	231
Zdarzenia	232
Stany	232
Komponenty	233
Tworzenie prototypów przy użyciu programów Sketch i Framer Studio	239
Prototyp prezentacji powitalnej	239
Prototyp szuflady nawigacyjnej	249
Prototyp dolnego paska nawigacyjnego	261
W praktyce	266
Bądź elementem społeczności	267
Materiały do nauki	267
Testowanie na telefonie	267
Bądź na bieżąco z nowościami o Framer Studio	267
Podsumowanie	268
Rozdział 9. Testowanie z udziałem użytkowników	269
Zrozumienie znaczenia testowania użyteczności	270
Określenie, co należy testować	271
Wybór odpowiedniej metody	273
Planowanie testu	276
Definiowanie celów	276
Scenariusze i zadania	277
Definiowanie skryptu	278
Przygotowanie środowiska	280

Przeprowadzanie testów użyteczności	281
Rekrutacja uczestników badania	282
Przedstawienie procesu	283
Obserwacja zachowań użytkowników	284
Identyfikacja kluczowych aspektów i podsumowanie wyników	285
W praktyce	286
Zapraszanie członków zespołu do oglądania sesji testowych	286
Nauka jest ważniejsza od potwierdzania	286
Sprawdzanie, czy odpowiadamy na pytania badawcze	287
Łączenie badań z pomiarami	287
Podsumowanie	288
Rozdział 10. Bibliografia i odwołania	289
Ogólne	289
Model Kano	289
Zasady Gestaltu	289
Informacje o modelu umysłowym	290
Informacje o modelu pojęciowym	290
Informacje o dokumentach person	290
Informacje o sortowaniu kart	290
Informacje o testowaniu drzewa (odwrotnym sortowaniu kart)	291
Informacje o diagramach pokrewieństwa	291
Informacje o metodzie szalonych ósemek	291
Informacje o ankietach	291
Informacje o podróżach użytkowników	292
Informacje o scenariuszach	292
Stopniowe ujawnianie	292
Efekt wzajemności	293
Leniwe logowanie	293
Testowanie przy pomocy użytkowników	293
Wzorce mobilne	293
Architektura informacji	294
Projekty wizualne	294
Prototypowanie	294
Tumult Hype	294
CoffeeScript	294
Framer	295
JavaScript	295
jQuery	295
Źródła obrazków	295
Rozdział 1.	295
Rozdział 2.	296
Rozdział 3.	296
Rozdział 4.	297
Rozdział 5.	297
Rozdział 6.	298
Rozdział 7.	298
Rozdział 8.	300
Rozdział 9.	300
Skorowidz	301

Prototypowanie — ożywianie pomysłów

Ja nie wyjaśniam, nie mówię, ja pokazuję.

— *Lew Tołstoj*

Przewidzenie sukcesu produktu nie jest prostym zadaniem. Historia ludzkości jest pełna zarówno produktów, które odniosły sukces, jak i tych, którym się to nie udało. Choć proste zabawki, takie jak hula-hoop, odniosły niebywały sukces, to jednak inne zabawki, jak na przykład Swing Wing (patrz rysunek 6.1) działający na podobnej zasadzie, w ogóle nie zyskały popularności.



Rysunek 6.1. Zabawka Swing Wing

Swing Wing to zabawka zaprojektowana, by bawić się poprzez szybkie ruchy głowy do przodu i do tyłu. Jeśli chodzi o popularność, okazała się wielką kląpą, choć zasada jej działania była podobna do hula-hoop, w przypadku którego zabawa polega na ruszaniu biodrami i które odniosło niezwykle sukces (powyższe zdjęcia pochodzą z [https://en.wikipedia.org/wiki/Swing_Wing_\(toy\)](https://en.wikipedia.org/wiki/Swing_Wing_(toy)), https://archive.org/details/swing_wing).

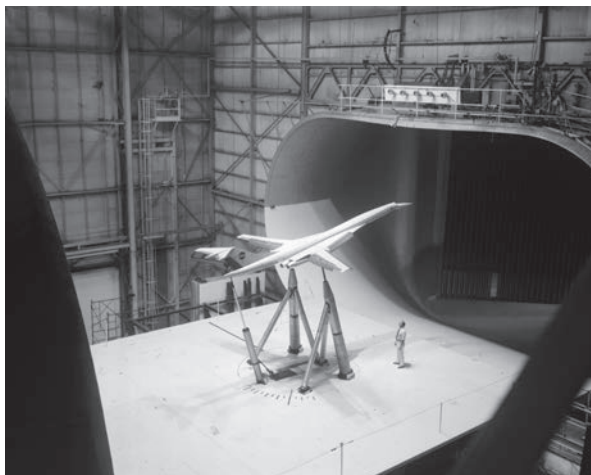
Każda aplikacja zaczyna się od pomysłu i nigdy nie ma pewności, czy dany pomysł się sprawdzi w praktyce, czy nie. Czekanie na stworzenie produktu tylko po to, by sprawdzić słusność pomysłu, stanowi poważne ryzyko — jeśli rzeczy nie pójdą po naszej myśli, to trudno już będzie zareagować i coś zmienić. Zatem bardzo przydatna byłaby możliwość znacznie wcześniejszego przekonania się, jak użytkownicy reagują na nasz produkt.

Na szczęście istnieje sposób na wcześniejszą ewaluację pomysłów — a wszystko dzięki prototypowaniu. Prototyp to symulacja wybranych części rozwiązania, która pozwala na poznanie opinii użytkowników na jego temat i ich reakcji jeszcze przed stworzeniem rzeczywistego produktu. W ten sposób nie trzeba czekać na dokończenie produktu, by się dowiedzieć, czy nasz pomysł zadziała zgodnie z oczekiwaniami. Prototypowanie pozwala oszczędzać czas i dostarczać lepszych produktów, które z większym prawdopodobieństwem zaspokoją rzeczywiste potrzeby użytkowników.

Tworzenie prototypów jest szybkie, lecz wymaga znajomości celów, w jakich prototypy są tworzone, doboru odpowiednich narzędzi i zaplanowania całego procesu. W tym rozdziale zamieścimy porady dotyczące wszystkich tych aspektów prototypowania, tak byśmy mogli ożywiać nasze pomysły możliwie jak najszybciej.

Z odpowiednim podejściem do prototypowania

Tworzenie prototypów jest powszechną praktyką w wielu dyscyplinach. Każdy rozumie zalety, jakie daje przetestowanie odpowiednio zmniejszonego modelu w tunelu aerodynamicznym, przed rozpoczęciem produkcji prawdziwego samolotu (patrz rysunek 6.2). Znacznie lepiej jest sprawdzić i wprowadzić odpowiednie modyfikacje w aerodynamice mniejszego modelu niż w czasie lotu już zbudowanego samolotu. Dokładnie to samo dotyczy aplikacji.



Rysunek 6.2. Model naddźwiękowego samolotu transportowego w pełnej skali testowany w tunelu aerodynamicznym w 1973 roku (źródło: https://commons.wikimedia.org/wiki/File:SST_model_in_Wind_Tunnel.jpeg)

Prototypy mogą wyglądać podobnie do prawdziwych produktów, lecz ich natura jest całkowicie odmienna. Prototyp nie musi działać, musi jedynie wyglądać tak, jakby działał. Co więcej, prototyp próbuje zasymulować jedynie kilka aspektów wybranych spośród wszystkich możliwości funkcjonalnych końcowego produktu — są to w szczególności te aspekty, których dokładniejsze zbadanie wydaje się projektantowi najważniejsze.

Prototypowanie wykorzystuje fakt, że *użytkownicy nie dbają o wewnętrzne szczegóły działania produktu*. Zwracają uwagę jedynie na jego widoczne aspekty. Zasymulowanie niektórych widocznych rozwiązań czasami w zupełności wystarcza do odtworzenia doświadczeń, jakie użytkownicy będą mieć podczas korzystania z gotowego produktu, i wyciągania na tej podstawie wniosków. Choć nasza rzeczywista aplikacja będzie musiała korzystać z bazy danych lub jakichś zewnętrznych usług, to prototyp może prezentować użytkownikom spreparowane informacje. Nikt nie zwróci uwagi na te różnice, choć przygotowanie ich będzie wymagało pewnego nakładu pracy. Jeśli użytkownik kliknie przycisk wyszukiwania w prototypie i na ekranie zostaną wyświetlone jakieś wyniki, to nie będzie on dociekał, skąd te informacje pochodzą.

A zatem w **prototypowaniu** chodzi o stworzenie pewnej iluzji jak najmniejszym nakładem pracy. Jeśli spróbujemy za jednym razem zawrzeć w prototypie zbyt wiele aspektów rozwiązania, wzrośnie poziom jego złożoności, a w pewnym momencie może się on nawet zacząć zbliżać do złożoności produktu końcowego. Dlatego prototypy powinny koncentrować się na konkretnym aspekcie i służyć do szybkiego zebrania informacji na jego temat.

Ograniczenie zakresu prototypu ma kluczowe znaczenie dla maksymalizacji korzyści, jakie nam on zapewni. W przypadku prototypu samolotu można koncentrować się bądź to na jego aerodynamice, bądź na projekcie wnętrza; lecz nie na obu tych aspektach jednocześnie. W przypadku sprawdzania aerodynamiki samolotu można by użyć modelu w odpowiedniej skali; natomiast pomysły projektu wnętrza można by prezentować w odpowiednio urządzonej sali. Stworzenie prototypu obejmującego jednocześnie oba te aspekty projektu byłoby jednak znacznie trudniejsze.

Dobór sposobu prototypowania najlepiej spełniającego nasze potrzeby

Podczas prototypowania bezustannie należy poszukiwać *równowagi pomiędzy realizmem i szybkością*. Z jednej strony chcemy bowiem, by prototypy w możliwie jak największym stopniu odtwarzały doświadczenia, jakie użytkownicy będą mieć w trakcie korzystania z finalnego produktu. Natomiast z drugiej strony dbałość o wszelkie szczegóły wymaga czasu, a nam zależy na jak najszybszym sprawdzeniu prawidłowości naszych pomysłów.

Idealny punkt równowagi pomiędzy realizmem i szybkością będzie inny dla każdego prototypu. Poniższe strategie mogą pomóc go znaleźć, a tym samym uzyskać jak największe korzyści z pracy włożonej w przygotowanie prototypów.

- Koncentrujemy się na konkretnym scenariuszu.
- Starajmy się upraszczać interakcje w zależności od ich znaczenia.
- Z góry określimy, czego chcemy się dowiedzieć.

Skoncentrowanie się na jednym scenariuszu pomaga upraszczać tworzony prototyp i zachować szybkie tempo pracy. Można zdecydować się na ograniczenie liczby opcji, jakie zostaną uwzględnione w prototypie, aby zmniejszyć jego złożoność, a jednocześnie by nie sprawiał on wrażenia zbyt niekompletnego i pozwalał na zapewnienie realistycznych doświadczeń.

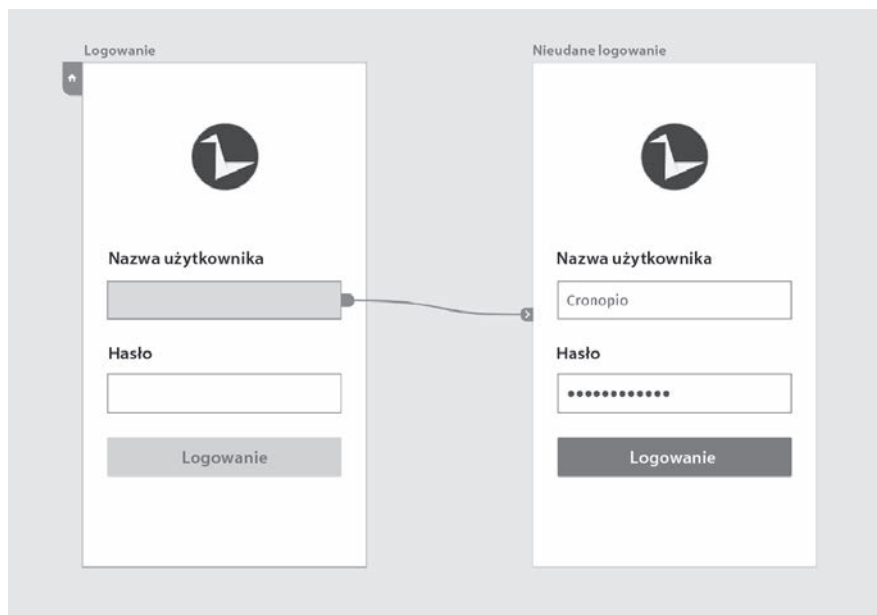
Tworzony prototyp odtwarzający konkretną reprezentatywną sytuację musi obsługiwać jedynie minimalną liczbę kroków, a my możemy z góry wybrać ten scenariusz. Na przykład w prototypie aplikacji turystycznej możemy pokazać w prezentowanych wynikach kilka miast, a jednocześnie pozwolić użytkownikom wyłącznie na wybór Paryża. Testując prototyp z udziałem użytkowników, możemy przedstawić scenariusz w taki sposób, by zachęcić ich do wybrania właśnie tego miejsca docelowego.

Zamiast obsługiwać wszystkie możliwe interakcje z użytkownikiem na tym samym poziomie szczegółowości, warto zastanowić się nad *uproszczeniem interakcji w zależności od ich znaczenia*. Każda wykonywana przez użytkownika interakcja z prototypem może być obsługiwana na różnym poziomie szczegółowości, zależnym od znaczenia danej interakcji. Na przykład możemy pominąć animowane efekty przejść pomiędzy poszczególnymi ekranami, jeśli zależy nam na poznaniu ogólnych wrażeń i opinii użytkowników na temat aplikacji. Pomimo braku tych efektów taki prosty prototyp w zupełności wystarczy, by przekonać się, czy użytkownicy uważają, że sposób nawigowania pomiędzy poszczególnymi ekranami aplikacji jest sensowny, czy nie. W przypadku badania konkretnych akcji, takich jak anulowanie powiadomień, można poświęcić nieco więcej czasu na wierne odtworzenie używanych gestów, animacji oraz innych aspektów interfejsu użytkownika, aby dokładniej i bardziej szczegółowo odtworzyć doświadczenia użytkownika.

Podczas takiego upraszczania interakcji można całkowicie pominąć pewne kroki pośrednie. Na przykład, jeśli tworzony prototyp zawiera ekran **Logowanie**, lecz nie stanowi on głównego celu badań, na których prototyp się koncentruje, to możemy przedstawić ten ekran w formie uproszczonej. Pokazanie ekranu logowania z już wypełnionymi polami jest może nieco mniej realistyczne, jednak powinno w zupełności wystarczyć użytkownikom do zrozumienia działania całego procesu (patrz rysunek 6.3).

Wiele osób nawet nie zauważy takiego przeskoku, na nieco podobnej zasadzie, na jakiej seria statycznych obrazów wyświetlanych w kinie jest przez nas postrzegana jako płynny ruch. W wielu przypadkach ludzki mózg jest w stanie wypełniać takie luki, a testy z wykorzystaniem użytkowników pokażą, ile kroków można pominąć, by iluzja ciągłości została zachowana.

I ostatnia sprawa: *konieczność zdefiniowania, czego chcemy się dowiedzieć, już na samym początku badań*. Tworzenie prototypu jest znacznie łatwiejsze, jeśli precyzyjnie zdefiniujemy jego cele, jeszcze zanim zaczniemy nad nim pracować. Określając, czego się chcemy dowiedzieć, możemy zadać sobie pytanie, czy wysiłek włożony w stworzenie konkretnej części prototypu pomoże w dotarciu do wyznaczonego celu. Na tej podstawie można określić koszty i korzyści oraz wybrać najprostszy typ prototypu, który pozwoli nam dowiedzieć się jak najwięcej.



Rysunek 6.3. Prototyp procesu logowania obejmujący dwa ekrany oraz jedną łączącą je interakcję przedstawiony w programie Adobe InDesign. Końcowy prototyp będzie początkowo prezentował pierwszy ekran, a następnie, kiedy użytkownik dotknie pola Nazwa użytkownika, zostanie wyświetlony drugi ekran z wypełnionymi polami tekstowymi symulującymi proces wpisywania danych uwierzytelniających

W zależności od celu prototypu można wybierać różne style oraz używać różnych narzędzi. Istnieje wiele różnych typów prototypów oraz wiele narzędzi do ich tworzenia. W kolejnych punktach rozdziału przedstawimy dostępne możliwości oraz sytuacje, w jakich warto je stosować.

Wybór typu prototypu

W zależności od przeznaczenia oraz czasu dostępnego na ich przygotowanie prototypy mogą być bardzo różne. Ilość potencjalnych typów prototypów, które można tworzyć, jest nieograniczona. W końcu w prototypowaniu chodzi o eksperymentowanie. Niemniej jednak istnieją pewne podejścia, które warto wykorzystać podczas szukania inspiracji do wypracowania własnego sposobu tworzenia prototypów. Te różne podejścia można scharakteryzować w oparciu o dwa kluczowe aspekty:

- **poziom wierności** — który determinuje, jak realistycznie prototyp wygląda;
- **poziom interakcji** — definiujący, jak prototyp reaguje na interakcje z użytkownikiem.

W zależności od poziomu wierności można znaleźć prototypy, które w różnym stopniu przypominają rzeczywiste produkty:

- **Prototypowanie na papierze** jest sposobem prototypowania o niskim poziomie wierności. Polega ono na wycinaniu, łączeniu oraz przesuwaniu kawałków papieru z narysowanymi na nich różnymi częściami interfejsu użytkownika (patrz rysunek 6.4). Można naszkicować różne ekrany aplikacji, pokazywać je użytkownikom i zmieniać aktualnie pokazywaną kartkę na inną, symulując w ten sposób interakcję, która nastąpiłaby, gdyby użytkownik dotknął jakiegoś elementu naszkicowanego na papierze.

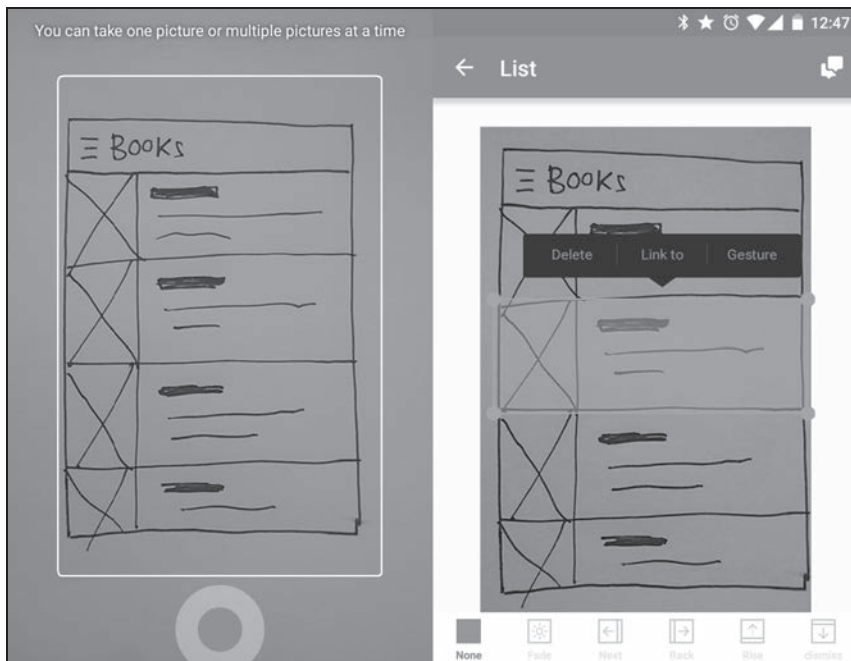


Rysunek 6.4. Prototyp papierowy, w którym różne elementy interfejsu użytkownika zostały połączone w formie kolażu (źródło: <https://www.flickr.com/photos/21218849@N03/7984460226/>)

W razie tworzenia takich prototypów można korzystać z przygotowanych szablonów, takich jak te dostępne na stronie <http://sneakpeekit.com>, aby rysować szkice na tle kształtu przypominającego ramkę przeglądarki. Można także wydrukować różne elementy interfejsu użytkownika, takie jak przyciski i pola wyboru, a następnie odpowiednio je łączyć. Elastyczność medium, jakim jest papier, pozwala błyskawicznie szkicować nowe stany w odpowiedzi na nieprzewidziane wcześniej akcje wykonywane przez użytkownika. Niemniej jednak ze względu na to, że użytkownicy prowadzą interakcje ze szkicami na papierze, znacznie trudniej im będzie w pełni wyobrazić sobie korzystanie z finalnego produktu.

Niski próg wejścia związany z korzystaniem z prototypów papierowych sprawia, że rozwiązanie to świetnie nadaje się do wykorzystania na wspólnych warsztatach projektowych, w trakcie których możemy zachęcać użytkowników do wyrażania swoich własnych pomysłów. Prototypy przygotowane przez użytkowników mogą nie zawierać pomysłów gotowych do natychmiastowego wykorzystania, jednak przydadzą się projektantom do odnajdywania problemów, które należy rozwiązać, oraz określania tych aspektów rozwiązania, na których użytkownikom najbardziej zależy.

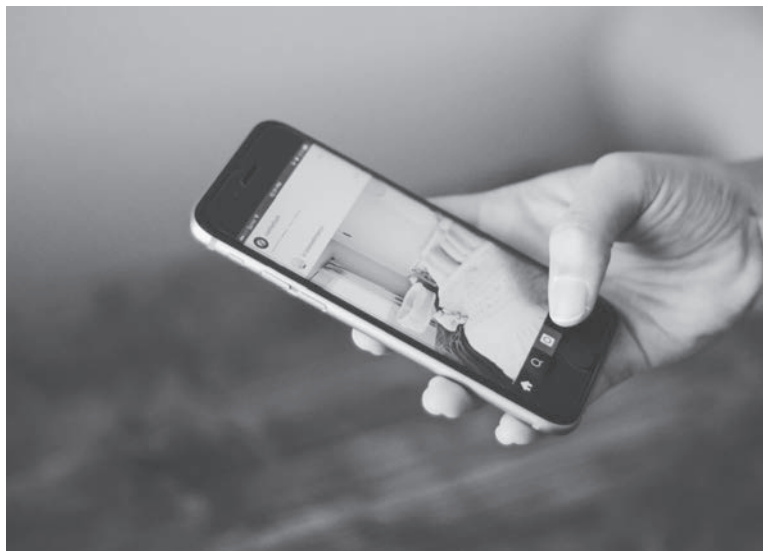
- **Prototypy cyfrowe niskiej wierności** także bazują na podobnym poście — łączeniu wielu szkiców w celu zasymulowania interakcji — jednak przenoszą go do świata cyfrowego (patrz rysunek 6.5). Takich prototypów można używać na rzeczywistych urządzeniach i dzielić się nimi cyfrowo ze znajomymi lub współpracownikami. Także w tym przypadku użytkownicy będą oglądać jedynie szkice stanowiące abstrakcyjną reprezentację produktu.



Rysunek 6.5. POP jest narzędziem do tworzenia prototypów działającym na urządzeniach mobilnych; pozwala ono na robienie zdjęć szkiców (z lewej) oraz definiowanie obszarów interaktywnych i powiązań pomiędzy nimi (z prawej)

Szkice można wykonać na papierze, a następnie zrobić ich zdjęcia, przy przeniesić je do formatu cyfrowego; można też w całości tworzyć je cyfrowo. W obu przypadkach szkice będą ze sobą łączone w aplikacji do prototypowania, pozwalającej określać sposoby reakcji na różne interakcje z użytkownikiem.

- **Prototypy o wysokiej wierności** są na tyle realistyczne, że nie sposób ich odróżnić od rzeczywistego produktu (patrz rysunek 6.6). Choć mogą one traktować pewne aspekty pobieżnie lub pomijać niektóre etapy, by zasymulować zamierzone doświadczenia, to jednak użytkownik będzie mógł w pełni je odczuć i zanurzyć się w nich. W tym przypadku użytkownicy będą komunikować swoje doznania w sposób bezpośredni, bez konieczności wyobrażania sobie czegokolwiek. Tworzenie takich prototypów wymaga bardziej zaawansowanych narzędzi, niezbędnych do przygotowania realistycznych materiałów i użycia ich w prototypie.



Rysunek 6.6. Czy to aplikacja, czy prototyp? Brak możliwości odróżnienia prototypu od aplikacji pozwala użytkownikom pogłębić doświadczenia

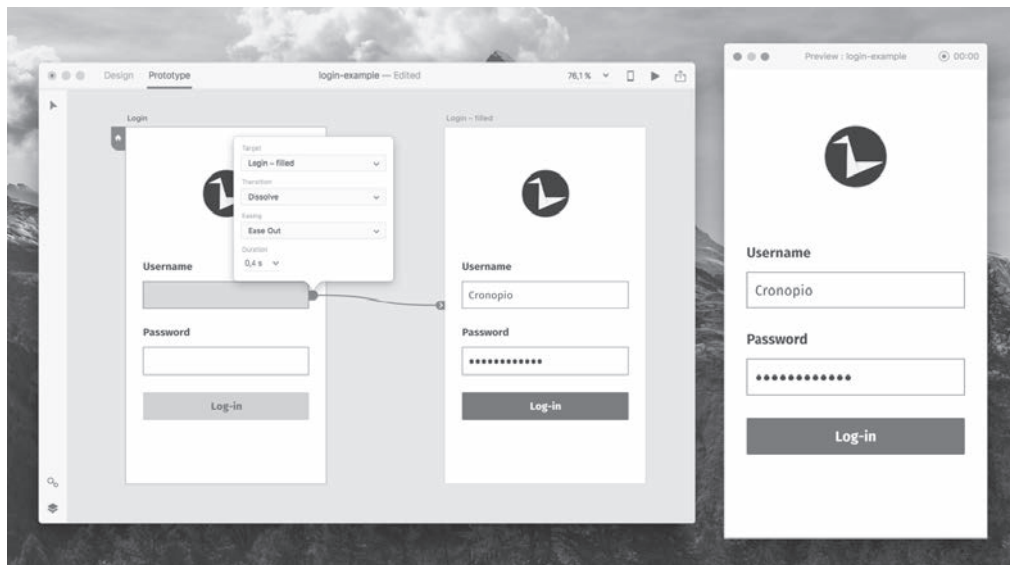
Kolejnym czynnikiem po poziomie wierności determinującym naturę prototypów jest ich **poziom interakcji**. Prototypy mogą udostępniać różne poziomy interakcji, określając tym samym, czy rola użytkownika będzie bardziej aktywna, czy pasywna.

- **Prototypy pozbawione interakcji** zawierają statyczne obrazy pokazujące, jak produkt będzie wyglądał w konkretnym kontekście, ewentualnie klipy wideo prezentujące postać interakcji. W takich prototypach użytkownicy nie mają kontroli, dzięki czemu unikamy konieczności przygotowywania prototypu na ewentualne interakcje z użytkownikiem. Osoby przeglądające taki prototyp mogą komentować to, co widzą, bądź też mówić, co sądzą na temat pomysłów odzwierciedlonych w prototypie. Prototypy tego typu doskonale nadają się do zbierania opinii od szerszych grup odbiorców, jednak nie zapewniają możliwości zbierania informacji dotyczących faktycznego działania.
- **Prototypy interaktywne** zapewniają użytkownikom możliwość interakcji z prototypem w taki sam sposób, w jaki później byłby używany finalny produkt. W porównaniu z prototypami pozbawionymi interakcji prototypy interaktywne zapewniają użytkownikom głębsze doświadczenia i pozwalają im zachowywać się w bardziej naturalny sposób. Umożliwiają one także gromadzenie wniosków w oparciu o faktyczne zachowania użytkowników, a nie jedynie na podstawie ich początkowych wrażeń i opinii.

W tej książce skoncentrujemy się głównie na interaktywnych prototypach o wysokiej wierności, gdyż pozwalają one na gromadzenie najbardziej wartościowych opinii. Niemniej jednak przedstawione techniki można wykorzystać podczas tworzenia prototypów dowolnych typów. Na przykład po utworzeniu prototypu interaktywnego bez trudu będzie można przygotować klip wideo prezentujący konkretny tok pracy, uzyskując w ten sposób prototyp pozbawiony interakcji.

Dobór narzędzi do tworzenia prototypów

Istnieje wiele różnych narzędzi służących do tworzenia prototypów (patrz rysunek 6.7), a w przyszłości najprawdopodobniej pojawi się ich jeszcze więcej. Niemniej jednak wiele z nich wykorzystuje podobne rozwiązania, dzięki czemu znajomość jednego z nich może się przydać także podczas korzystania z innych.



Rysunek 6.7. Program Adobe Experience Design jest narzędziem do projektowania i prototypowania pozwalającym na wizualne łączenie różnych stron poprzez tworzenie przejść. Symulator pokazany z prawej strony prezentuje jeden widok w danej chwili, pozwalając użytkownikowi na interakcje z prototypem (źródło: zrzut ekranu z programu Adobe Experience Design)

Narzędzia bazujące na **przechodzeniu pomiędzy stronami** korzystają z różnych stron oraz obszarów interaktywnych. Stronę można zdefiniować w taki sposób, by zawierała konkretny widok aplikacji. Strony te mogą być tworzone w oparciu o pojedynczy obraz interfejsu użytkownika zaprojektowany w zupełnie innym narzędziu lub składać się z połączenia różnych kontrolki interfejsu użytkownika.

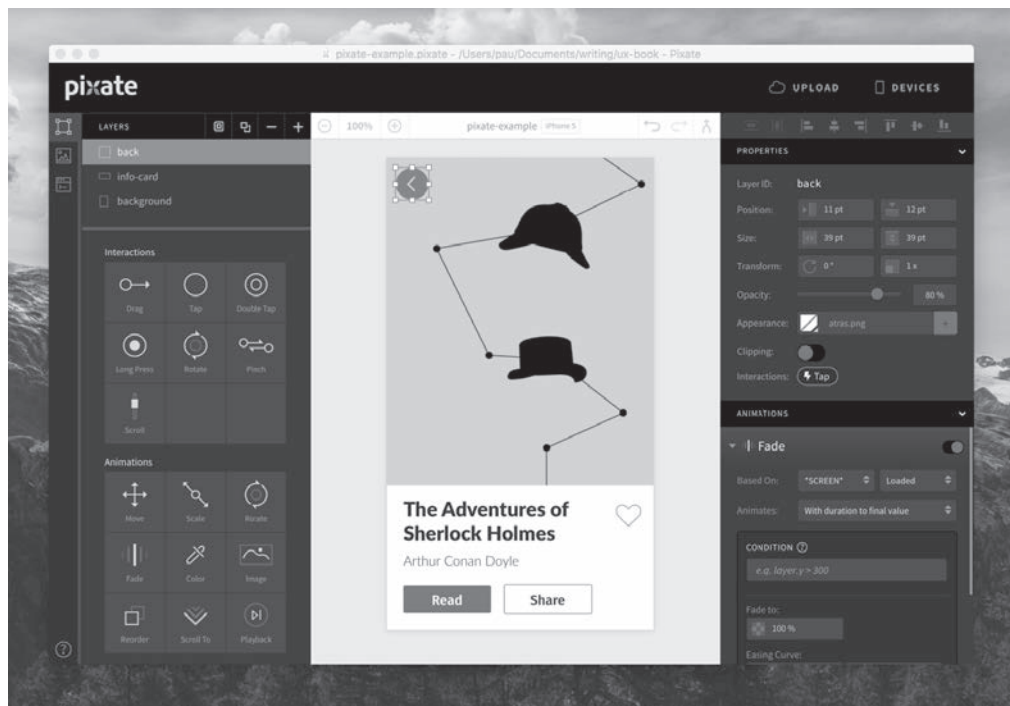
W celu tworzenia interakcji możemy definiować na stronie interaktywne obszary i przejścia do innych stron. W ten sposób możemy określić, że dotknięcie obszaru ikony menu spowoduje przejście do widoku menu. Oprócz odwzorowań sposobów poruszania się po aplikacji można także definiować przejścia symulujące zmiany stanów, na przykład zmianę pustego pola wyboru na pole zaznaczone.

Często występującym ograniczeniem narzędzi tego typu są problemy z reprezentacją wielu równoczesnych stanów, wynikające z konieczności dostosowania stanu prototypu do ścieżki użytkownika realizującego różne interakcje. Innymi słowy, jeśli prototyp pozwala na wykonanie opcjonalnej czynności, której efekty muszą zostać w jakiś sposób zachowane, takiej jak

usunięcie elementu z listy, to może się okazać, że konieczne będzie powielenie wielu widoków, by uwzględnić sytuację, gdy element został usunięty oraz gdy nie został usunięty z listy.

Narzędzia korzystające z **interakcji na warstwach** pozwalają na tworzenie prototypów poprzez niezależne manipulowanie różnymi elementami. W takim narzędziu można stworzyć prototyp zawierający listę elementów, a następnie niezależnie kontrolować widoczność każdego z nich.

Takie rozwiązanie zapewnia pełną kontrolę nad statusem poszczególnych elementów prototypu. Przykład programu tego typu został przedstawiony na rysunku 6.8.

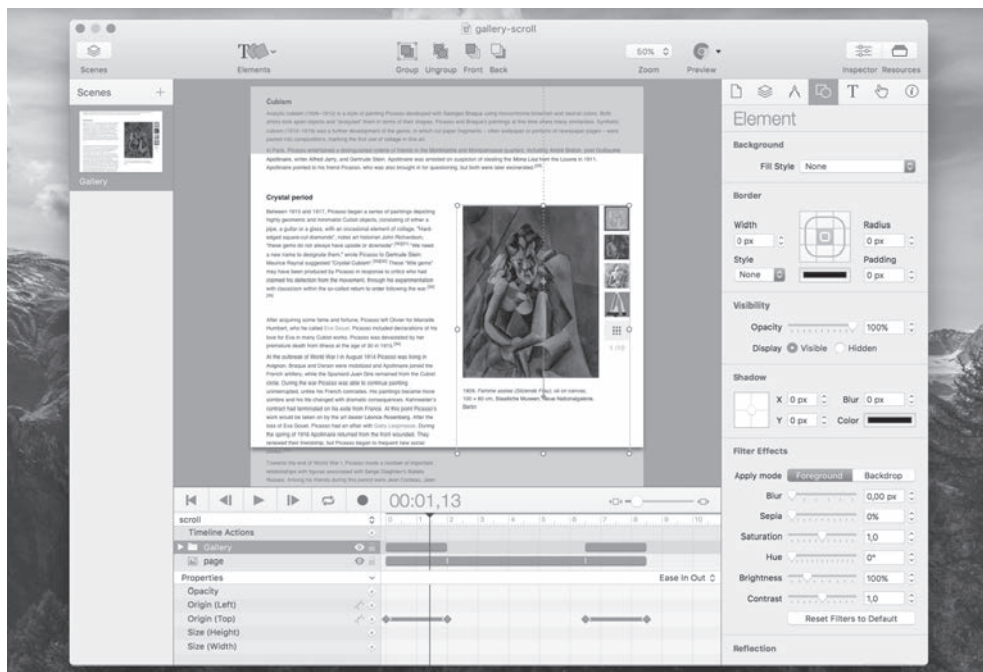


Rysunek 6.8. Program Pixate korzysta z warstw (wyświetlonych po lewej) i pozwala na modyfikowanie różnych właściwości (po prawej), w tym na definiowanie dla każdej z warstw odrębnych interakcji i animacji (źródło: zrzut ekranu z programu Pixate)

Częstą wadą narzędzi tego typu jest to, że zmuszają one do wydzielania fragmentów i tworzenia prototypów z wielu części. Ponieważ szkielety i atrapy zazwyczaj są tworzone w odrębnych narzędziach, zmusza to nas do eksportowania odpowiednich fragmentów z narzędzia używanego do tworzenia projektu graficznego i importowania ich w narzędziu do prototypowania. Niektóre programy udostępniają zaawansowane narzędzia do importu, które upraszczają i skracają ten proces.

Narzędzia działające w oparciu o oś czasu także korzystają z warstw, jednak w ich przypadku kluczowe znaczenie ma upływ czasu. Dzięki wykorzystaniu osi czasu zapewniają one dokładną kontrolę nad animacjami i efektami przejść stosowanymi w prototypach.

W prototypach tworzonych w narzędziach tego typu można stosować zasady projektowania ruchu i dostosowywać szczegóły wszystkich animacji. Przykład programu tego typu został przedstawiony na rysunku 6.9.

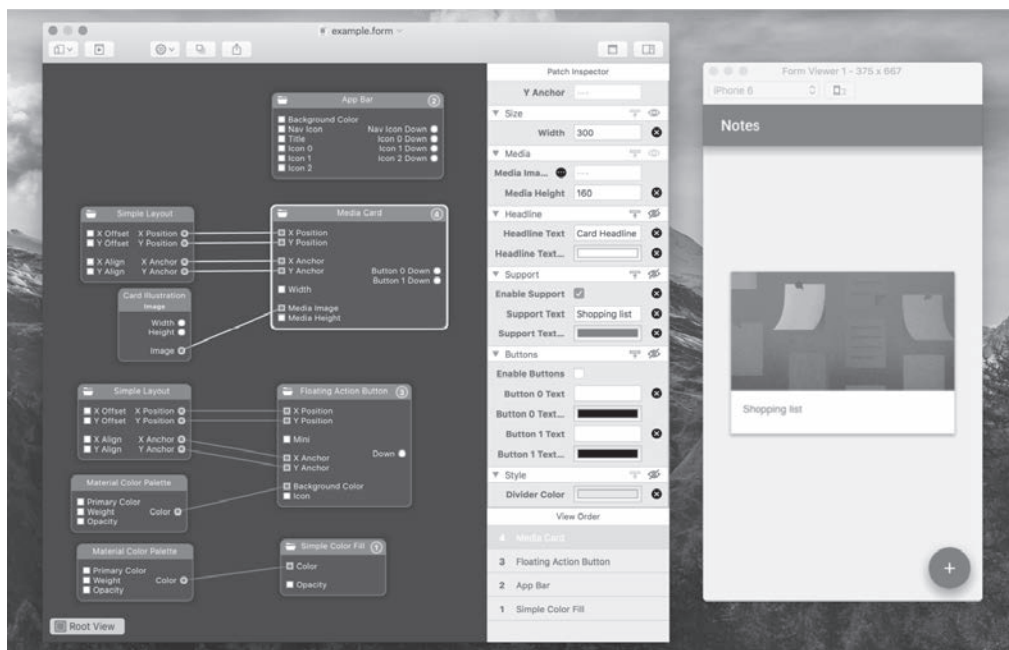


Rysunek 6.9. Program Tumult Hype korzysta z osi czasu wyświetlonej u dołu, by pokazywać, jak różne elementy i ich właściwości zmieniają się wraz z upływem czasu (źródło: zrzut ekranu z programu Tumult Hype)

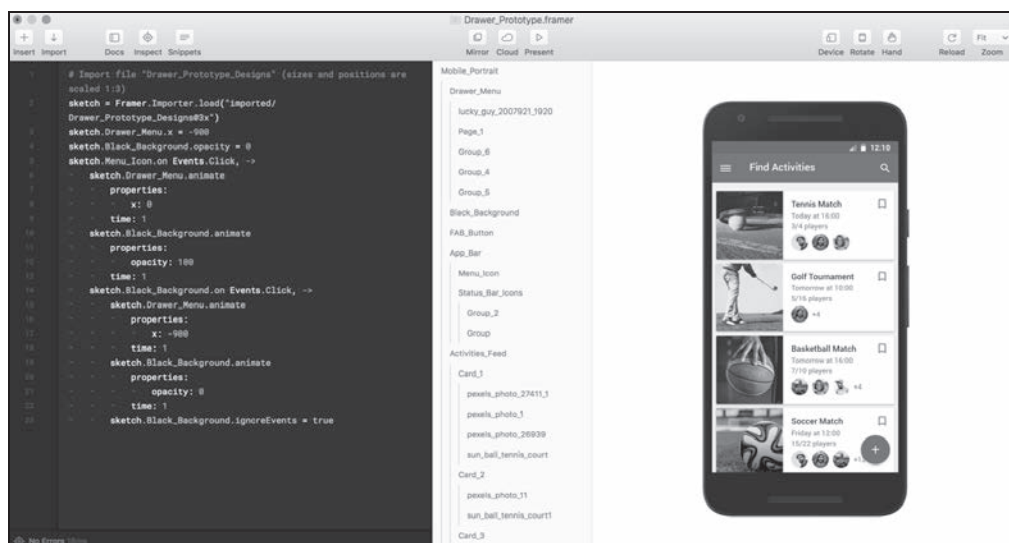
Narzędzia wykorzystujące **połączenia pomiędzy komponentami** działają w oparciu o paradygmat programowania wizualnego. Pozwalają one na łączenie bloków reprezentujących interaktywne komponenty, interakcje użytkownika oraz elementy logiczne. Przykład takiego programu został przedstawiony na rysunku 6.10.

Działanie prototypu możemy definiować poprzez łączenie bloków. Blok reprezentujący gest uszczypnięcia możemy połączyć z właściwością określającą wielkość obrazka, symulując w ten sposób operację jego zmniejszania i powiększania. Definiując takie połączenia, można także stosować pewne bloki logiczne, pozwalające na przykład na ograniczenie zakresu powiększania lub kontrolę szybkości efektu przejścia.

Narzędzia wymagające pisania kodu pozwalają na definiowanie prototypów przy użyciu języka programowania (patrz rysunek 6.11). Wykorzystują one pewne zasady prototypowania, by ułatwić tworzenie przejść pomiędzy stanami i definiowanie animacji oraz by uprościć zapisywanie ich w formie kodu. Narzędzia tego typu często pozwalają także na integrację z innymi komponentami programowymi, zapewniając tym samym dużą elastyczność. Niemniej jednak ta elastyczność może kusić, by dodawać do prototypu niepotrzebne szczegóły. Zatem stosując takie narzędzia, musimy starać się pamiętać o zasadach prototypowania i dbać, by tworzone prototypy koncentrowały się na wybranych zagadnieniach.



Rysunek 6.10. Form to narzędzie do tworzenia prototypów, którego działanie opiera się na zasadzie łączenia komponentów. Z lewej strony widoczny jest edytor pozwalający na dodawanie bloków i łączenie ich właściwości; z prawej — symulator prezentujący wynikowy prototyp (źródło: zrzut ekranu z programu Form)



Rysunek 6.11. Framer to narzędzie do tworzenia prototypów wymagające pisania kodu. Warstwy, ich właściwości oraz wszystkie interakcje muszą być definiowane w postaci tekstowej. Do wizualizacji prototypu używane są dodatkowe narzędzia (źródło: zrzut ekranu z programu Framer)

Często pojawia się konieczność utworzenia prototypu pozbawionego interakcji w oparciu o prototyp interaktywny. Do tego celu można użyć narzędzi do tworzenia treści multimedialnych. Na przykład programu do przechwytywania ekranu można użyć do zarejestrowania interakcji z prototypem, a programu do edycji wideo — do przycięcia zarejestrowanego materiału, zmontowania go, dodania ścieżki dźwiękowej i dopracowania finalnego klipu.

Planowanie prototypu

Prototypowanie to proces poznawczy i nie powinniśmy obawiać się popełniania błędów, o ile tylko na ich podstawie możemy się czegoś dowiedzieć. Nie ma nic złego w odkrywaniu pytań dotyczących projektu, na które jeszcze nie znamy odpowiedzi, i odpowiadaniu na nie podczas tworzenia prototypów. Częścią tego procesu jest eksperymentowanie. Oczywiście trochę planowania pomoże nam zaoszczędzić nieco czasu.

Podczas tworzenia prototypów zachowanie efektywności pracy wymaga chodzenia na skróty i traktowania niektórych aspektów rozwiązania pobieżnie. Takie decyzje mogą ograniczyć liczbę dostępnych opcji i warto postarać się, choćby w pewnym stopniu, oszacować ich konsekwencje. W końcu nikt nie chciałby znaleźć się w sytuacji, w której w połowie prac nad prototypem okazuje się, że wybrane narzędzie nie obsługuje interakcji absolutnie niezbędnej dla kluczowej części prototypu.

W tym podrozdziale przedstawimy kilka zagadnień, które należy przemyśleć przez rozpoczęciem prac nad prototypem. Warto poświęcić nieco czasu na planowanie, niemniej w trakcie tworzenia prototypu można dowolnie eksperymentować.

Określanie, co ma zostać uwzględnione w prototypie

Znajomość celów, jakim ma służyć prototyp, ułatwi podejmowanie kluczowych decyzji podczas procesu jego tworzenia. Rozpoczynając prace nad prototypem, należy mieć jasność co do następujących zagadnień:

- **Cel:** czego chcemy się dowiedzieć dzięki tworzonemu prototypowi?
- **Odbiorcy:** od kogo chcemy zbierać informacje?
- **Scenariusz:** jakie aktywności w konkretnym kontekście będą obsługiwane przez prototyp?

Odpowiedzi na te pytania pozwolą nam wybrać sposób tworzenia prototypu oraz narzędzia, których można do tego użyć.

Należy bardzo precyzyjnie *określić cel, jakiemu ma służyć prototyp*. Na przykład możemy chcieć dowiedzieć się czegoś na temat ogólnego zrozumienia naszego pomysłu, przejrzystości systemu nawigacyjnego, łatwości odnajdywania konkretnych aktywności, płynności realizacji konkretnych zadań czy też wielu innych czynników mających wpływ na doświadczenia użytkownika. Jeden prototyp może być tworzony w kilku celach, jednak należy przy tym

unikając celów ogólnych, takich jak „czy pomysł spełni oczekiwania”. W przeciwnym razie może się okazać, że trudno będzie określić, które aspekty rozwiązania należy uwzględnić w prototypie, by móc odpowiedzieć na postawione pytania.

Musimy *pomóc odbiorcom w dostarczaniu nam użytecznych odpowiedzi*. Warto zastanowić się, czy wybrana grupa osób będzie w stanie odpowiedzieć na postawione pytanie oraz jak prototyp może im w tym pomóc. Jeśli tworzymy prototyp, by przedyskutować pomysły z innymi projektantami, to prototyp o niskiej wierności w zupełności wystarczy do tego celu; z kolei prototyp o dużej wierności lepiej spełni zadanie, kiedy będziemy chcieli przeanalizować konkretne szczegóły zachowań rzeczywistych użytkowników.

Prototyp powinien opierać się na kontekście, który jest znany odbiorcom, gdyż w takim przypadku łatwiej im będzie zrozumieć sytuację, w jakiej zostali postawieni. Jeśli tworzymy prototyp aplikacji wspomagającej podróżowanie, to zapewne jako przykładowe miejsce docelowe lepiej będzie wybrać znane miasto niż tajemniczą miejscowość o nazwie trudnej do wymówienia — oczywiście gdy właśnie takiej sytuacji nie chcemy przetestować.

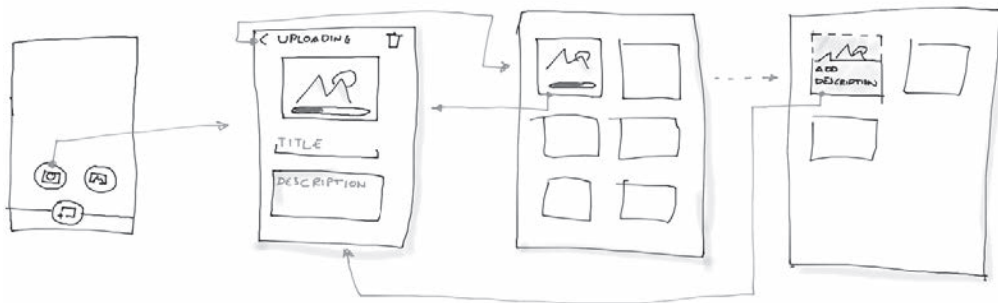
Trzeba się także zastanowić, w jaki sposób prototyp zostanie dostarczony wybranym odbiorcom. W zależności od ich liczby i lokalizacji można im dostarczyć urządzenia z zainstalowanym prototypem lub udostępnić im go, by mogli zainstalować go na własnych urządzeniach. Wiele narzędzi do prototypowania pozwala generować prototypy w formie kodu HTML, co dzięki powszechnej dostępności przeglądarek WWW sprawia, że z powodzeniem można je testować na wielu urządzeniach. Jednak istnieją także narzędzia tworzące prototypy przeznaczone dla konkretnych platform, co ogranicza liczbę potencjalnych odbiorców oraz zmusza ich do przeprowadzenia jakiegoś procesu instalacyjnego.

W zależności od wybranego scenariusza można także wskazać aspekty o różnej ważności — minimalny zestaw aspektów obsługiwanych przez dany scenariusz, aspekty wymagane w rozszerzonym kontekście oraz te, których nie trzeba uwzględniać w prototypie. Wysiłki należy skoncentrować na tych najważniejszych aspektach.

W niektórych przypadkach może być trudno precyzyjnie określić, co należy uwzględnić w prototypie. Możemy nie wiedzieć, które kierunki tworzenia projektu są najbardziej obiecujące. W takich sytuacjach można przygotować wiele prototypów i przeanalizować, jak użytkownicy reagują na poszczególne pomysły. Jednak ze względu na ograniczoną ilość czasu na przygotowanie prototypu konieczne będzie dokonanie wyboru pomiędzy tworzeniem jednego szczegółowego prototypu a kilku mniej szczegółowych.

Wybór sposobu tworzenia prototypu

Po wybraniu scenariusza, który zostanie zobrazowany w formie prototypu, konieczne będzie przedstawienie szczegółów wszystkich interakcji, które użytkownik będzie musiał wykonać w ramach tego scenariusza. Poszczególne kroki, które będzie miał uwzględniać prototyp, można przedstawić w formie **sekwencji szkiców** (ang. *sketchflow*). Oprócz samych szkiców reprezentujących poszczególne kroki zawiera ona także łączące je interakcje (patrz rysunek 6.12).



Rysunek 6.12. Sekwencja szkiców reprezentująca proces aktualizacji grafiki przeznaczonej do użycia w aplikacji do tworzenia prototypów

W tworzeniu własnych prototypów może pomóc przygotowanie sobie gotowych szablonów. Takie szablony przydadzą się także podczas przeglądania prac i oceniania ich postępów. Ponieważ prototypy mają sprawiać wrażenie, że działają, łatwo jest pominąć w nich pewne interakcje; przygotowane wcześniej szablony pozwalają bardziej systematycznie sprawdzać zawartość prototypów.

Nasz scenariusz może mieć idealną ścieżkę reprezentującą czynności, które użytkownik wykonałby w idealnej sytuacji. Jednak w razie tworzenia prototypów interaktywnych nie jesteśmy w stanie kontrolować działań użytkowników, a oni niejednokrotnie nie będą postępować zgodnie z naszą idealną ścieżką. Dlatego warto uwzględnić przejścia pozwalające użytkownikom cofnąć się oraz postarać się przewidzieć miejsca, w których użytkownicy mogą zbacać z idealnej ścieżki.

Zapewnienie użytkownikom realistycznego kontekstu pozwoli im lepiej pojąć i odczuć doświadczenia towarzyszące tworzeniu prototypu. Warto unikać zamienników takich jak „Lorem Ipsum” oraz używać sensownych i dobrze dobranych przykładów. W prototypie można uwzględnić także dodatkowe kroki, jako wstęp do dodatnia bardziej rozbudowanego kontekstu. Na przykład, jeśli chce się wzmocnić wrażenia z pierwszego uruchomienia aplikacji, to zamiast rozpoczynać prototyp od pokazania jej ekranu początkowego, można zacząć od przedstawienia ekranu sklepu z aplikacjami, z poziomu którego użytkownik ją zainstaluje.

W praktyce

Prototypowanie jest bardzo ważnym elementem procesu projektowania. Jednak nie wszyscy wiedzą, jaką rolę ono powinno pełnić. Oto kilka uwag, które mogą pomóc w prawidłowym wykorzystaniu prototypowania w praktyce.

Prototyp to nie wczesna wersja produktu

Termin „prototyp” jest często używany w wielu różnych znaczeniach. Zdarza się, że terminem tym jest określana wczesna wersja produktu, gdyż zawiera ona jeszcze sporo rzeczy wymagających dopracowania. Jednak takie niedopracowane produkty nie są w stanie w pełni

zaoferować finalnych doświadczeń użytkowników (gdyż nie są kompletne), ich przygotowanie wymaga dodatkowego wysiłku, a one same stanowią swoiste zobowiązanie do rozwoju produktu w określonym kierunku.

Prototypy tworzy się wyłącznie w celu przetestowania pomysłu, a brak w nich pewnych elementów jest efektem świadomych decyzji dotyczących ważności elementów prezentowanych użytkownikom oraz mających istotne znaczenie dla uzyskania odpowiedzi na postawione pytania dotyczące projektu. Prototyp udaje finalny produkt w tak dużym stopniu, jak to możliwe, aż do punktu, w którym jego ponowne wykorzystanie staje się bezcelowe. Nie jest to jednak problemem, gdyż największą wartością prototypów jest nauka i informacje; gdy te zostaną zdobyte, prototyp można porzucić.

Prototyp jest wart tysiąca spotkań

„Jeśli obraz jest wart tysiąca słów, to prototyp jest wart tysiąca spotkań”

— *słynne powiedzenie z firmy IDEO.*

Spotkania często są wypełnione dyskusjami opierającymi się na opiniach i spekulacjach. Materiały często używane na spotkaniach, takie jak konkretne hasła i statyczne obrazy, wciąż wymagają dużego umysłowego wysiłku, koniecznego do wyobrażenia sobie, jak pomysły będą się sprawdzać w praktyce.

Prezentując prototyp, należy się skoncentrować na opowieści, dzięki której odbiorcy wyobrażą sobie kontekst sytuacji. Koniecznie należy przy tym podkreślić problemy, które nasz projekt rozwiązuje, a zrezygnować z opisywania jego konkretnych cech. Prototyp może pełnić rolę okna do przyszłości, pokazującego, jak tworzony produkt będzie działać. Pozwala on na bezpośrednie doświadczenie rozwiązania i pomaga skoncentrować się na podanym kontekście.

Podsumowanie

W tym rozdziale przedstawiliśmy podstawowe zasady prototypowania. Podchodząc do tego z odpowiednim nastawieniem, można szybko nauczyć się przekształcania dowolnych pomysłów w coś rzeczywistego i pozyskiwania na tej podstawie cennych informacji. W ten sposób można błyskawicznie badać bardziej innowacyjne rozwiązania dla tworzonych aplikacji i poprawiać je na bazie opinii zebranych od rzeczywistych użytkowników.

Prototypowanie jest jednym z kluczowych etapów procesu projektowania. Jednak są osoby, które mogą postrzegać je jako etap opcjonalny. Zrozumienie zasad prototypowania i opanowanie różnych technik tworzenia prototypów pozwoli nam pokazać jego pełne możliwości i wykazać, że w rzeczywistości pozwala ono oszczędzać czas.

W kolejnych rozdziałach przedstawimy konkretne narzędzia ilustrujące poszczególne techniki prototypowania. Konkretnie rzecz biorąc, w następnym rozdziale opiszemy program Hype, dysponujący wspaniałymi możliwościami tworzenia animacji i pozwalający na koordynowanie ruchu poszczególnych elementów przyszłej aplikacji w celu pokazania, jak będzie ona działać.

Skorowidz

A

adaptacja, 99
adnotacje, 80
afordancje, 28
akcje drugorzędne, 116
analiza
 krytyczna, 90
 możliwości funkcjonalnych,
 71
 słownictwa, 71
Android, 125
animacje, 205, 231, 247
ankiety, 58, 291
architektura informacji, 56, 294
aspekty nieistotne, 37
automatyzacja, 148

B

badania, 38, 47, 86
 ewaluacyjne, 273
 generacyjne, 273
 podłużne, 275
badanie
 pomysłów, 77
 różnych rozwiązań, 129
 użytkownika, 43
bezpieczne poszukiwania, 34
bepośrednia obsługa ręczna,
31
blokowanie, 173
błędy, 34, 116
burza mózgów, 49

C

cechy ekranów, 126
cele, 41, 66, 93
ciemne wzorce, 41
cienie, 80
CoffeeScript, 221, 294
 funkcje, 224
 konstrukcje warunkowe,
 225
 metody, 227
 obiektów, 227
 operatory, 222
 pętle, 225
 typy danych, 224
 zmiennych, 222
czcionki, 129

D

definiowanie skryptu, 278
diagram pokrewieństwa, 48, 52,
291
dobór sposobu prototypowania,
159
dokumenty
 badawcze, 62
 person, 63, 290
dostarczanie inteligentnych
wartości domyślnych, 30
doświadczenia
 użytkownika, UX, 18, 40
 w domenę zagadnienia, 66

działanie

 pola wyszukiwania, 256
 prototypu, 246
 szuflady nawigacyjnej, 253

E

edytor
 kodu, 180
 sceny, 177
efekt
 animacji, 249
 obserwatora, 273
 wzajemności, 293
efekty
 czas trwania, 139
 szybkość, 140
ekrany, 109, 126
eksperymentowanie, 79
elastyczne wprowadzanie
danych, 30
elastyczność, 99
elementy
 interfejsu użytkownika, 127
 przeznaczone do dotykania,
 148

F

format
 GIF, 150
 JPEG, 150
 PDF, 150
 SVG, 150

formularze, 61
 Framer.js, 219, 228, 267, 295
 animacje, 231
 komponenty, 233
 stany, 232
 warstwy, 229
 zdarzenia, 232
 frameworki do tworzenia
 interfejsu, 101
 frustracje, 67
 funkcje, 224

G

galerie z miniaturkami, 112
 generowanie pomysłów, 79
 Gestalt, 289
 gesty, 135, 201
 gęstości pikseli, 143
 grupowanie, 24, 26
 pomysłów, 50
 grupy użytkowników, 73

H

humanizacja formularzy, 61
 Hype Tumbult
 edytor kodu, 180
 edytor sceny, 177
 importowanie mediów, 180
 inspektor właściwości, 177
 lista scen, 177
 łączenie animacji, 205
 obsługa
 gestów, 201
 nawigacji, 195
 przewijania, 195
 wprowadzania danych,
 211
 organizacja warstw, 217
 oś czasu, 178
 panel Actions, 185
 panel sceny, 187
 prototypowanie nawigacji,
 181
 przeglądanie efektów, 218
 reprezentacja stanu, 189
 stosowanie odrębnych
 warstw, 192
 warstwy, 178, 188

I

identyfikacja konfliktów, 87
 ikona, 128, 151
 aplikacji, 111
 iluzja introspekcji, 269
 importowanie mediów, 180
 informacje demograficzne, 64
 informowanie użytkowników,
 31
 inspektor właściwości, 177
 inspiracje, 93
 interakcje, 164
 jednokrotnego dotknięcia,
 114
 operujące na warstwach,
 166, 188
 użytkownika, 114
 interfejs użytkownika, 101,
 127
 iOS, 125

J

JavaScript, 295
 jednostki, 143
 język CoffeeScript, 221
 jQuery, 295

K

karty u góry aplikacji, 106
 kodowanie, 154
 kolor, 24
 komentarz, 115
 komponent, 233
 PageComponent, 236
 RangeSlider, 238
 ScrollComponent, 235, 252
 SliderComponent, 237
 TextLayer, 233
 kompromis, 94
 komunikacja, 32, 79
 komunikaty, 120
 konstrukcje warunkowe, 225
 koordynowanie animacji, 247
 kształty, 23
 o zaokrąglonych
 wierzchołkach, 25
 proste, 80

L

leniwe logowanie, 293
 liczby, 224
 limit pamięci roboczej, 57
 lista scen, 177

Ł

łańcuchy, 224
 łączenie
 animacji, 205
 badań z pomiarami, 287

M

makiety, 141
 marnowanie czasu
 użytkownika, 32
 menu, 127
 dolne, 106
 metafory, 29
 metoda szalonych ósemek, 84,
 291
 metody, 227
 badawcze, 47
 behawioralne, 273
 jakościowe, 273
 metryki, 288
 miniaturka, 209
 model
 Kano, 289
 pojęciowy, 46, 290
 umysłowy, 32, 44, 290
 modelowanie zachowań, 27
 modyfikowanie poziomu
 ważności, 37
 motywacje, 67

N

nakładki, 118
 narzędzia
 do szkicowania, 81
 do tworzenia prototypów, 165
 nawigacja, 104, 108, 109, 181,
 195
 nazwy
 grup, 51
 obszarów, 56

O

obciążenie poznawcze, 27
 obiekty, 224, 227
 obiektywizm, 74
 obrazy, 149
 obserwacja, 47
 obsługa
 błędów, 116
 gestów, 201
 wprowadzania danych, 211
 zdarzeń, 252
 obszar kompozycji, 244
 oczekiwania, 27
 odroczone logowanie, 122
 odświeżanie, 123
 okna dialogowe, 111
 operatory, 222
 opis, 66
 optymalizowanie, 37
 organizacja
 badań, 86
 treści, 56
 oś czasu, 178
 względna, 200

P

panel sceny, 187
 pasek nawigacyjny, 263
 pętle, 225
 piksele, 143
 niezależne od gęstości, 145
 skalowalne, 145
 piórko cyfrowe, 82
 planowanie
 długoterminowe, 59
 krótkoterminowe, 59
 projektu ogólnego, 38
 prototypu, 169
 testu, 276
 plik specyfikacji, 146
 pływające przyciski, 110
 podejście skoncentrowane
 na użytkowniku, 20
 odpowiedź w kształcie kółka,
 111
 podróz użytkownika, 63, 69,
 292
 pole wyszukiwania, 256

połączenia pomiędzy
 komponentami, 167
 pomiary użyteczności, 275
 porady, 118
 postrzeganie kształtów, 24
 potrzeby, 23, 27, 34
 konieczne, 36
 liniowe, 36
 sprzeczne, 37
 ukryte, 36
 powiadomienia, 110
 poziom interakcji, 164
 prawo bliskości, 26
 program
 Framer Generator, 220
 Framer Studio, 219, 239
 Sketch, 239
 Tumult Hype, 167, 173, 176
 progresywne aplikacje
 internetowe, 98
 projektowanie, 39
 w oparciu o kolumny, 102
 projekty wizualne, 294
 prostota, 37
 prototypowanie, 38, 157, 294
 na papierze, 162
 nawigacji, 181
 z wykorzystaniem kodu, 38,
 219
 z wykorzystaniem ruchu, 38,
 173
 złożonych zachowań, 205
 prototypy
 cyfrowe niskiej wierności,
 163
 dobór narzędzi, 165
 dolnego paska
 nawigacyjnego, 261
 działanie, 246
 interaktywne, 164
 o wysokiej wierności, 163
 planowanie, 169
 pozbawione interakcji, 164
 prezentacji powitalnej, 239
 program
 Framer Studio, 219, 239
 Hype, 176
 Sketch, 239
 sposoby, 159
 szuflady nawigacyjnej, 249,
 253

tworzenie, 158, 170
 wybór typu, 161
 znaczenie ruchu, 175
 przechodzenie pomiędzy
 stronami, 165
 przeciągnięcie w dół, 123
 przejścia, 136
 przestrzeń projektu, 78
 przycisk
 FAB, 252
 Wstecz, 108
 przyciski fizyczne, 126
 psychologia
 koloru, 24
 kształtu, 23
 pytania, 60
 badawcze, 287
 otwarte, 60
 wielokrotnego wyboru, 60

R

recenzja, 115
 redukcja czynników
 rozpraszających, 33
 rekrutacja uczestników badań,
 73
 reprezentacja stanu, 189
 responsywność, 99
 rozkład problemu, 86
 rozwiązania hybrydowe, 101
 ruch, 175

S

scenariusz, 63, 68, 277, 292
 scenorys, 89, 174
 sceny, 177
 sekwencja szkieletów, 170
 sesje analizy krytycznej, 90
 skeumorficzny interfejs, 45
 skeumorfizm, 45
 skrypt testu, 278
 sortowanie kart, 52, 290
 odwrotne, 55
 wariant otwarty, 53
 wariant zamknięty, 54
 specyfikacja, 143
 możliwości funkcjonalnych,
 132

stany, 232
 obszarów kompozycji, 242
 stopniowe ujawnianie, 122, 292
 stosowanie elastycznego
 wprowadzania danych, 30
 strony
 adaptacyjne, 100
 płynne, 99
 strzałki, 112
 szkicowanie, 77–80
 szkielety, 132

Ś

śledzenie ruchów oka, 276

T

tablice, 224
 tekst, 80
 testowanie, 38, 154, 269
 drzewa, 52, 55, 291
 makiet, 142
 na telefonie, 267
 przy pomocy
 użytkowników, 293
 użyteczności, 270
 moderowane, 274
 niemoderowane, 275
 testy
 A/B, 287
 definiowanie celów, 276
 definiowanie skryptu, 278
 obserwacja zachowań
 użytkowników, 284
 oglądanie sesji, 286
 planowanie, 276
 podsumowanie wyników,
 285
 przedstawienie procesu,
 283
 przygotowanie środowiska,
 280
 rekrutacja uczestników
 badania, 282
 scenariusze, 277
 użyteczności, 281
 zadania, 277
 tła, 149
 trajektoria, 136

trendy, 154
 Tumult Hype, 173, 294
 tworzenie
 aplikacji, 99
 interfejsu użytkownika, 101
 podgrup, 50
 podróży użytkownika, 69
 prototypów, 158, 165, 170,
 239
 scenarysów, 89
 typy danych, 224

U

udostępnianie, 115
 układ, 134
 ekranu, 113
 umieszczanie narzędzi, 32
 unikanie
 błędów, 34
 ślepych uliczek, 34
 uszczegółowienie rozwiązania,
 38
 UX, user experience, 18, 40
 użyteczność, 270
 użytkownicy, 40
 docelowi, 72
 nieobsługiwani, 72
 obsługiwani, 72

W

warstwy, 178, 188, 192, 229
 wartości logiczne, 224
 widok kalendarza, 198
 wiele ekranów, 109
 wybór typu prototypu, 161
 wydajność, 32
 wygląd, 127
 wyjaśnianie poprzez
 porównanie, 32
 wymaganie minimalnego
 zaangażowania, 30
 wyrażanie pomysłów, 89
 wysuwana szuflada, 104
 wyszukiwanie, 124, 258, 260
 wyświetlanie komunikatu, 120
 wytyczne projektowe, 29
 wywiady, 47
 wzajemność, 125

wzorze
 mobilne, 38, 293
 projektowe, 103
 interfejsu użytkownika,
 104, 121
 zachowań, 122

Z

zadania, 277
 zależności czasowe, 137
 zapewnienie poczucia kontroli,
 31
 zapory drogowe, 119
 zasady
 Gestaltu, 26, 289
 minimalnego zaskoczenia,
 27
 projektowe, 17, 22
 zasoby, 143
 zdarzenia, 232, 252
 zdjęcie, 65
 zmienne, 222
 znaczenie ruchu, 175
 znajdowanie pomysłów, 38

PROGRAM PARTNERSKI

— GRUPY HELION —

1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA
Helion 

Zrób na użytkownika najlepsze wrażenie!

User experience (UX), czyli wrażenie użytkownika, w ogromnym stopniu decyduje o sukcesie produktu cyfrowego. Przed przystąpieniem do właściwego pisania aplikacji warto zastosować wywodzące się z UX techniki analizy rzeczywistych potrzeb użytkowników. W ten sposób łatwiej jest tworzyć produkty, z których będą oni korzystać z przyjemnością. Odpowiadanie na potrzeby odbiorcy — poprzez staranne projektowanie interakcji z użytkownikiem — jest szczególnie istotne w przypadku aplikacji mobilnych. UX pozwala znacznie obniżyć ryzyko porażki i zaoszczędzić czas podczas projektowania.

Ta książka jest praktycznym wprowadzeniem do projektowania efektywnych aplikacji mobilnych. Zaproponowano tu techniki odpowiednie dla wczesnych etapów rozwoju aplikacji, pozwalające na skrócenie czasu prac projektowych i zmniejszenie ryzyka. Sporo uwagi poświęcono testowaniu pomysłów, ich wyborowi i dopracowywaniu. Opisano techniki korzystania z prototypów o niskiej i wysokiej wierności z wykorzystaniem profesjonalnego oprogramowania narzędziowego, a także wyjaśniono wady i zalety ich poszczególnych rodzajów. Przedstawiono podstawowe sposoby i etapy procesu testowania. Co istotne, zaprezentowane w książce idee i techniki można zastosować niezależnie od wielkości budżetu przeznaczanego na projekt.

Najważniejsze zagadnienia:

- zasady projektowania i budowy scenariuszy
- badanie potrzeb użytkowników
- korzystanie ze szkiców, wzorców mobilnych, szkieletów i makiet
- narzędzia do prototypowania: Tumult Hype i Framer Studio
- testowanie prototypu z udziałem użytkowników

Pablo Perea jest entuzjastą technologii, programowania i inżynierii. Prowadzi badania, tworzy prototypy, projektuje aplikacje i testuje nowe rozwiązania. Znakomicie zna i rozumie wiele różnych języków programowania. Jest osobą elastyczną, łatwo się adaptuje i lubi poznawać nowe kultury.

Dr Pau Giner jest projektantem. Lubi poznawać oraz prototypować nowe pomysły i zawsze ma pióro w zasięgu ręki. Pracuje dla Wikimedia Foundation oraz dla innych otwartych projektów związanych z udostępnianiem wiedzy innym ludziom. Organizował warsztaty dla projektantów i opublikował wiele artykułów.

Helion 	<i>Sprawdź nasze szkolenia!</i>	KOD KORZYŚCI Sięgnij po więcej! ▶	
 helion.pl	 AKADEMIA IT & BUSINESS WWW.SZKOLENIA.HELION.PL	ISBN 978-83-283-4699-4	
 0 801 339900		9 788328 346994	
 0 601 339900	INFORMATYKA W NAJLEPSZYM WYDANIU	Cena: 59,00 zł	

Packt