

# Analiza porównawcza narzędzi do budowy prototypów interfejsów

Stanisław Lipski\*, dr inż. Marek Miłośz

Politechnika Lubelska, Instytut Informatyki, Nadbystrzycka 36B, 20-618 Lublin, Polska

**Streszczenie.** Analizę porównawczą narzędzi do prototypowania interfejsów użytkownika przeprowadzono przy pomocy metody mieszanej polegającej na realizacji aktywnego eksperymentu z wykorzystaniem narzędzi na grupie badawczej, przeprowadzeniu badań ankietowych oraz analizie eksperckiej parametrów technicznych. Rezultaty badań pozwoliły wyznaczyć wartości zmiennych diagnostycznych, które to użyte w metodzie selekcji wielokryterialnej posłużyły do wyboru najlepszego narzędzia.

**Słowa kluczowe:** interfejs użytkownika; prototypowanie; narzędzia wspomagające; wybór wielokryterialny

\* Autor do korespondencji.

Adresy E-mail: staslipski@interia.pl, m.milosz@pollub.pl

## Comparative analysis tools for interfaces prototyping

Stanisław Lipski\*, Marek Miłośz

Institute of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

**Abstract.** Comparative analysis tools for user interfaces prototyping was conducted using a mixed method consisting in the implementation of active experiment with the use of tools for the research group, conducting surveys and expert analysis of technical parameters of tools for prototyping. Study results have helped determine the values of diagnostic variables that have been used in the multi-selection method of the best tool selection.

**Keywords:** user interface; prototyping; supporting tools; multi-criteria selection

\* Corresponding author.

E-mail addresses: staslipski@interia.pl, m.milosz@pollub.pl

### 1. Wstęp

Istotnym elementem w cyklu wytwarzania oprogramowania, wpływającym na końcowy rezultat odgrywa prototypowanie. Jest ono stosowane w początkowej fazie wytwarzania oprogramowania, które stanowi dopełnienie dokumentacji ukazującej zakres wymagań projektu. Zastosowanie prototypów w fazie projektowania ma na celu umożliwienie projektantom testowanie nowych koncepcji dotyczących konstrukcji systemu. Oceniając prototyp, projektanci mogą identyfikować wymagania funkcjonalne, problemy z użytecznością i wydajnością, które mogą być rozpatrywane przed fazą realizacji. Prototypowanie szczególnie rekomendowane jest do projektowania interfejsów użytkownika. Sposób przekazania wymagań do GUI (ang. *Graphical User Interface*) w postaci tekstowej nie jest wystarczający, co może doprowadzić do błędnej interpretacji przez zespół projektowy. Połączenie prototypowania ze współpracą z użytkownikiem końcowym może być efektywnym sposobem projektowania graficznych interfejsów użytkownika [2].

Dokonując podziału prototypów, można uwzględnić kryteria obejmujące zakres projektowanego systemu.

**Prototypem poziomym** (ang. *Horizontal Prototype*) nazywa się prototyp, który ukazuje szeroki zakres działania systemu z uwzględnieniem tylko istotnych elementów. Budowa tego rodzaju prototypu ma na celu zobrazowanie wymagania przyszłego systemu i ułatwić zatwierdzenie

specyfikacji projektu. Do głównych zalet tego prototypu zaliczyć można lepsze ukazanie potrzeb użytkownika oraz rzeczową ilustrację specyfikacji wymagań [1, 13].

**Prototypowanie pionowe** (ang. *Vertical Prototype*) jest to rodzaj szczegółowego prototypu, który uwzględnia jedynie pewne aspekty całego projektu. Stworzenie prototypu pionowego wybranego fragmentu systemu, może posłużyć kontroli wybranej architektury przy założeniu pewnych scenariuszy testowych. Prototypowanie pionowe wykorzystane może być w początkowej fazie projektowania jak również na etapie kreowania różnych wariantów aplikacji [3].

Prototypowanie może opierać się na dwóch wariantach związanych z porzuceniem prototypu lub jego dalszą ewaluacją. **Porzucenie prototypu** (ang. *Throwaway Prototype*) polega na stworzeniu modelu, który ostatecznie zostanie porzucony i nie będzie dalej rozwijany. Przygotowanie tego typu prototypu służy głównie do ukazania wymagań lub do podjęcia decyzji wyboru odpowiedniej wersji systemu [2, 4]. Inną metodą jest **prototypowanie ewolucyjne** (ang. *Evolutionary Prototyping*) mające na celu stworzenie dokładnego prototypu, który stopniowo będzie udoskonalany aż do produktu finalnego. Podstawowym czynnikiem jest współpraca z użytkownikami końcowymi, którzy dostarczają kluczowych informacji dotyczących zmian w przyszłym systemie [5, 10, 14].

## 2. Metoda selekcji wielokryterialnej

Wybierając odpowiednie narzędzie należy przeanalizować wiele czynników, które wpływają na wybór tego odpowiedniego. Przeprowadzając analizę porównawczą badane są pewne cechy produktów pod względem efektywności czy przydatności danych narzędzi w konkretnych projektach. Uzyskane wyniki pomogą zaobserwować różnice pomiędzy poszczególnymi narzędziami i dokonać odpowiedniego wyboru.

Analiza wielokryterialna wykorzystywana jest przy wspomaganiu procesu decyzyjnego. Stosowana do oceny i wyboru najkorzystniejszego rozwiązania wśród analizowanych wariantów. Przed przystąpieniem do analizy należy określić zmienne diagnostyczne, czyli zestaw zmiennych przedstawiających charakterystykę wybranych obiektów. Dobór odpowiednich zmiennych diagnostycznych w znacznym stopniu wpływa na wyniki badania, a nieodpowiednia selekcja może negatywnie wpłynąć na końcowe rezultaty. Wyodrębnienie zmiennych diagnostycznych dla narzędzi do prototypowania odbywa się w tym przypadku przy pomocy kryteriów poza statystycznych, czyli merytorycznych i formalnych. Wybierane są te cechy, które dla badanego przedmiotu odgrywają istotną rolę do przeprowadzenia analizy porównawczej [12].

W analizie porównawczej zmienne (kryteria) można traktować w sposób współmierny lub zastosować wobec nich odpowiednie wagi różnicowe, których wyodrębniona wartość stanowi o ważności danego kryterium. Przy nadawaniu wag muszą być spełnione dwa kryteria:

- wagi zdefiniowane, jako nieujemne,
- suma wartości zmiennych diagnostycznych powinna być równa jeden

Dokonując wielowymiarowej analizy porównawczej na wybranych zmiennych diagnostycznych wiąże ze sobą konieczność przeprowadzenia transformacji w celu ujednoczenia obserwowanych parametrów. Proces polegający na zmianie parametrów nazywany jest transformacją normalizacyjną. Jednym z typów normalizacji jest unitaryzacja, polegająca na uzyskaniu zmiennych o znormalizowanym zakresie przedstawionym przez różnicę pomiędzy wartościami maksymalnymi i minimalnymi. Wynikiem otrzymanej wartości jest zmienna w przedziale [0;1]. Ogólną formułę normalizacji można przedstawić wzorem:

$$Z_{ij} = \frac{X_{ij} - \min\{X_{ij}\}}{\max\{X_{ij}\} - \min\{X_{ij}\}} \quad (1)$$

gdzie:  $X_{ij}$  – zaobserwowana wartość j-tej zmiennej w i-tym obiekcie,  $\min\{X_{ij}\}$  – minimalna wartość j-tej zmiennej w i-tym obiekcie,  $\max\{X_{ij}\}$  – maksymalna wartość j-tej zmiennej w i - tym obiekcie

Po przeprowadzeniu normalizacji poszczególnych zmiennych można przystąpić do uporządkowania zbioru

obiektów opartej na metodzie bez wzorcowej. Polega na dokonaniu uśrednienia unormowanych wcześniej wartości zmiennych przypisując im odpowiednie wagi [9]. Do wyznaczenia zmiennej syntetycznej uwzględnia się następujący wzór:

$$S_i = \sum_{j=1}^m Z_{ij} W_j \quad (2)$$

gdzie:  $Z_{ij}$  – znormalizowana wartość zmiennej i-tym obiekcie w j-tej zmiennej,  $W_j$  – waga kryterium j-tej zmiennej.

## 3. Narzędzia informatyczne wspomagające prototypowanie interfejsu

Przy budowaniu prototypu interfejsu można skorzystać z szeregu dostępnych narzędzi dzięki którym w łatwy sposób można stworzyć makietę systemu. Posłużyć można się uniwersalnymi narzędziami takimi jak Microsoft PowerPoint lub Microsoft Visio. Istnieją również narzędzia dedykowane wyłącznie do budowy prototypu interfejsu. Różnią się między sobą liczbą dostępnych komponentów, stopniem odzwierciedlenia czy rodzajem oferowanej licencji [2, 11]. Do przeprowadzenia analizy porównawczej wybrane zostały trzy narzędzia komputerowe wspomagające prototypowanie.

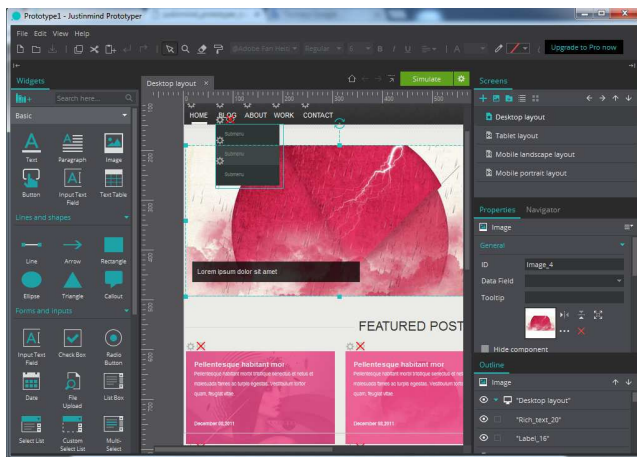
### 3.1. Narzędzie Justinmind

Justinmind to oprogramowanie do tworzenia interaktywnych prototypów stron internetowych i aplikacji mobilnych. Prezentowane narzędzie udostępnia szeroki zakres gotowych komponentów, dzięki którym możliwe jest zbudowanie najbardziej wymagającego projektu, bez znajomości języków programowania. Konstruowanie interfejsu odbywa się poprzez przeciąganie komponentów i umieszczanie ich na wybranym ekranie. Każdy z gotowych elementów posiada możliwość edycji i dostosowania go do indywidualnych potrzeb jak również zastosowanie interakcji pomiędzy poszczególnymi modułami. Stworzony interfejs można w łatwy sposób przetestować i przekazać innym osobom zaangażowanym w projekt. W przypadku aplikacji mobilnych możliwe jest przetestowanie prototypu na dowolnym urządzeniu przy pomocy dedykowanej aplikacji na systemy operacyjne Android i IOS. Podczas projektowania zespołowego możliwa jest jednoczesna praca wielu osób nad tym samym projektem z funkcją dodania oceny i komentarza dla prezentowanych komponentów. Narzędzie oferuje również wygenerowanie szczegółowej dokumentacji technicznej wykonanego projektu i wyeksportowania go do dowolnego formatu [6]. Wygląd głównego okna programu przedstawiony został na rys. 1.

### 3.2. Narzędzie Pidoco

Kolejne prezentowane narzędzie do prototypowania interfejsów działa w modelu chmury obliczeniowej SaaS (ang. *Software as a Service*) i nie wymaga pobierania ani instalacji na stacjach roboczych. Do korzystania

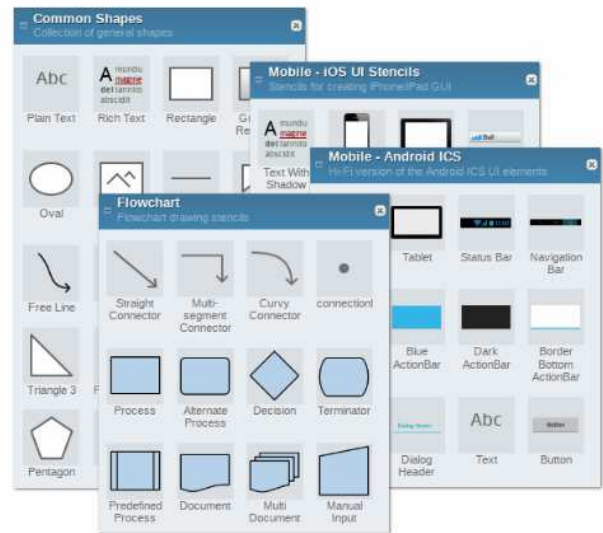
z oprogramowania wymagany jest dostęp do sieci Internet i przeglądarka internetowa. Narzędzie oferuje możliwość planowania, projektowania i testowania stron internetowych, aplikacji mobilnych i aplikacji biznesowych. System umożliwia użytkownikom tworzenie szkieletów makiet i interaktywnych prototypów interfejsów niskiej jakości bez konieczności programowania. Kreowanie interfejsów w danym narzędziu bazuje na przeciąganiu komponentów z dostępnych zasobów i umieszczaniu ich w tworzonym prototypie – rys. 2. Wobec każdego elementu można zastosować predefiniowaną interakcję i umożliwić użytkownikom możliwość poruszania się pomiędzy poszczególnymi ekranami systemu [7].



Rys. 1 Okno projektowe narzędzia Justinmind

### 3.3. Narzędzie Pencil Project

Pencil Project służy tak jak poprzednie narzędzia do tworzenia makiet stron internetowych, aplikacji mobilnych czy do projektowania rozbudowanych systemów. Narzędzie to jest programem typu desktop i rozpowszechniane jest na licencji typu Freeware. Oprogramowanie można pobrać ze strony producenta i zainstalować na komputerze z systemami operacyjnymi Windows, Os X lub Linux. Pencil Project zapewnia dostęp do różnych kolekcji elementów interfejsu użytkownika, dedykowane są również dla systemów mobilnych Android czy IOS [8]. Na rys. 3 przedstawione są komponenty do wykorzystania w programie Pencil Project.



Rys. 3 Dostępne komponenty programu Pencil Project

## 4. Problem badawczy i plan badań

Problemem poruszonym w danej pracy jest zagadnienie prototypowania i dedykowane narzędzia do budowania prototypów interfejsów. Każde z narzędzi posiada swój indywidualny zestaw cech, które mogą mieć wpływ na cały proces prototypowania i efekt finalny. Dokonując analizy porównawczej, można zweryfikować przydatność poszczególnych narzędzi i wybrać odpowiednie oprogramowanie, które sprosta wymaganiom projektu.

W pracy postawiona została następująca teza:

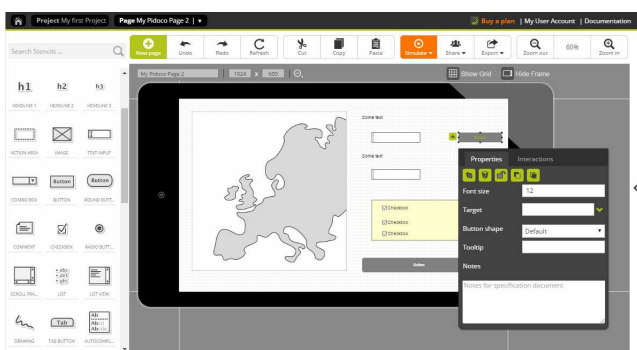
*Istnieje możliwość wyboru optymalnego narzędzia do budowy prototypu interfejsu.*

Do przedstawionych problemów zostały również sformułowane następujące hipotezy:

- H1. Możliwe jest sformułowanie jednoznacznego kryterium do porównania efektywności narzędzia do budowy prototypów.
- H2. Różne narzędzia do budowy prototypów interfejsów mają zróżnicowaną efektywność.

### 4.1. Przebieg badań

Badanie obejmowało przeprowadzenie pewnego doświadczenia na grupie ludzi, którego celem było zaobserwowanie efektywności wybranych narzędzi, przy budowie zdefiniowanego prototypu systemu. Za jednym z kryteriów efektywności narzędzia uwzględniony został łączny czas zaznajomienia się z danym narzędziem jak również zbudowania prototypu interfejsu. Przed przystąpieniem do badań zdefiniowana została dokumentacja systemu na podstawie, której musiał zostać stworzony model systemu. Dokumentacja została przekazana osobom zaangażowanym w doświadczenie. Każde z narzędzi było testowane przez trzy osoby biorące udział w badaniu. Po zapoznaniu się z danym narzędziem i zbudowaniu prototypu



Rys. 2 Główne okno narzędzia Pidoco

dotąd każdy uczestnik badania musiał wypełnić ankietę, składającą się z pewnej liczby pytań, w której miał dokonać oceny danego narzędzia pod względem przyjazności interfejsu, ilości dostępnych komponentów czy stabilności pracy.

Badanie uwzględniało następujące elementy:

- określenie kryterium analizy porównawczej dla narzędzi do budowy prototypów,
  - określenie parametrów technicznych,
  - wybór parametrów efektywnego prototypowania,
- nadanie wybranym kryteriom odpowiednich wag
- przedstawienie czasów budowy prototypu uzyskanych przez członków grupy badawczej,
- prezentacja wyników ankiet,
- przeprowadzenie normalizacji danych z ankiety przy pomocy wzoru (1),
- przypisanie poszczególnym cechom narzędzi odpowiedniej ilości punktów na podstawie własnych obserwacji i danych uzyskanych z ankiety,
- przedstawienie zsumowanych wyników dla poszczególnych narzędzi przy pomocy wzoru (2),
- ocena uzyskanych wyników analizy.

#### 4.2. Charakterystyka grupy badawczej

Grupa badawcza składała się z dziewięciu osób, która miała za zadanie stworzyć prototyp interfejsu użytkownika w przydzielonym im narzędziu i zmierzyć całkowity czas budowy prototypu systemu. Aby z przeprowadzonych badań można było uzyskać wiarygodne wyniki, a analizę przeprowadzić w sposób rzetelny wybrano grupę badawczą składającą się z absolwentów uczelni wyższych na kierunku informatyka.

Osoby biorące udział w doświadczeniu, nie znały wcześniej przydzielonego im narzędzia, dzięki czemu była możliwość zbadania całkowitego czasu nauki i tworzenia prototypu.

#### 4.3. Specyfikacja projektu

Do zweryfikowania efektywności wybranych narzędzi sporządzona została dokumentacja projektu z wymaganiami systemu. Projekt obejmował stworzenie prototypu aplikacji mobilnej pod system Android lub iOS, która umożliwi komunikację tekstową pomiędzy osobami, rejestrowanymi i zalogowanymi do systemu.

Wymagania aplikacji mobilnej zostały zdefiniowane w następujący sposób:

- osoba, która chce korzystać z aplikacji musi przejść proces rejestracji użytkownika, podając takie dane jak login, hasło, e-mail, rok urodzenia, i zdjęcie profilowe;
- logowanie do aplikacji mobilnej odbywa się poprzez wprowadzenie przez użytkownika loginu i hasła zadeklarowanych podczas rejestracji;
- użytkownik, który zapomniał swojego loginu lub hasła ma możliwość przywrócenia swoich danych poprzez wprowadzenie podanego podczas rejestracji adresu e-mail, na który wysłany zostanie login z nowym hasłem;
- zalogowany użytkownik ma możliwość nawiązania rozmowy tekstowej z innymi osobami, które dokonały

rejestracji, poprzez zastosowanie wyszukiwania przy wprowadzeniu odpowiedniego kryterium (login, przedział wiekowy) lub skorzysta z dodatkowej funkcji losowego wyszukiwania kontaktów;

- użytkownicy mają możliwość wysyłania pomiędzy sobą wiadomości tekstowych;
- osoba korzystająca z aplikacji mobilnej dysponuje możliwością tworzenia własnej grupy znajomych.

Wymagania prototypu są następujące:

- zbudowany prototyp musi wykazywać pełną interaktywność w taki sposób, aby osoby testujące prototyp aplikacji mogły poruszać się pomiędzy poszczególnymi ekranami systemu;
- wykorzystać dostępne komponenty i zastosować wobec nich odpowiednią interakcję i dynamikę w ramach testowanego narzędzia.

#### 4.4. Dobór zmiennych diagnostycznych

Dobierając parametry porównawcze brano były pod uwagę czynniki, które mogą wpływać na właściwy dobór oprogramowania do budowania prototypów interfejsu użytkownika dla konkretnego projektu jak również wpływ na efektywność procesu prototypowania. Nie wszystkie parametry są tak samo ważne i przy porównywaniu należy zastosować odpowiednie wagi, które ukazą, jakie parametry są istotne, a które w mniejszym stopniu wpłyną na wyniki badania. Parametry zostały podzielone na dwie kategorie. W pierwszej zostały uwzględnione parametry techniczne oprogramowania zaprezentowane w tab. 1.

Tabela 1. Parametry techniczne oprogramowania

Nazwa kryterium (waga)	Wskaźniki	Przypisane wagi
Uruchomienie na systemach operacyjnych (0,3)	wszystkie	1 pkt.
	Windows	0,50 pkt.
	Mac Os X	0,50 pkt.
Budowanie prototypów na platformy (0,2)	wszystkie	1 pkt.
	Android	0,25 pkt.
	iOS	0,25 pkt.
	Web	0,25 pkt.
Odzwierciedlenie interfejsu (0,15)	Duże	0,50 pkt.
	Średnie	0,30 pkt.
	Małe	0,20 pkt.
Eksport prototypu (0,2)	wszystkie	1 pkt.
	HTML	0,33 pkt.
	Aplikacja mobilna	0,34 pkt.
	PDF, .DOC, .PNG	0,33 pkt.
Cena narzędzia (0,15)	Wersja komercyjna	0,12 pkt.
	Wersja bezpłatna	0,70 pkt.
	Wersja próbna	0,18 pkt.

Druga grupa kryteriów odnosi się już do samego procesu prototypowania. Uwzględnione zostały tu czynniki, które realnie wpływają na szybkość wykonanego prototypu. Najważniejszym czynnikiem, który otrzymał najwyższą wagę jest czas wykonania prototypu, który zależy w głównej mierze od łatwości użytkowania i przyswojenia danego narzędzia. Niższy czas budowania prototypu jest lepiej punktowany. Kryteria zostały przedstawione w tab. 2.

Tabela 2. Parametry efektywności

Nazwa kryterium (waga)	Wskaźniki	Przypisane wagi
Czas budowy prototypu (dane uzyskane z eksperymentu) (0,45)	Najniższy czas	0,75 pkt.
	Średni czas	0,20 pkt.
	Najwyższy czas	0,05 pkt.
Wsparcie (poradniki, dokumentacja) (0,05)	Duże	0,75 pkt.
	Średnie	0,15 pkt.
	Małe	0,10 pkt.
	Brak	0 pkt.
Testowanie na urządzeniach mobilnych (0,1)	Tak	1 pkt.
	Nie	0 pkt.
Interaktywność prototypu (0,05)	Duża	0,75 pkt.
	Średnia	0,15 pkt.
	Niska	0,10 pkt.
Stopień trudności budowania prototypu (dane uzyskane z ankiety) (0,1)	Niski (1-2)	0,75 pkt.
	Średni (2-4)	0,15 pkt.
	Wysoki (4-5)	0,10 pkt.
Stabilność pracy narzędzia (dane uzyskane z ankiety) (0,1)	Duża (4-5)	0,75 pkt.
	Średnia (2-4)	0,15 pkt.
	Niska (1-2)	0,10 pkt.
Szybkość działania programu (dane uzyskane z ankiety) (0,1)	Duża (4-5)	0,75 pkt.
	Średnia (2-4)	0,15 pkt.
	Niska (1-2)	0,10 pkt.
Liczba dostępnych komponentów (dane uzyskane z ankiety) (0,05)	Duża (3-4)	0,75 pkt.
	Średnia (2-3)	0,15 pkt.
	Niska (1-2)	0,10 pkt.

## 5. Rezultaty badań

Z przeprowadzonych badań uzyskano prezentowane w tab. 3 czasy budowy prototypów interfejsów dla poszczególnych narzędzi.

Tabela 3. Czasy budowy prototypów

Czas w godzinach	Narzędzia do prototypowania		
	Pencil Project	Justinmind	Pidoco
t1	1:22	1:55	0:16
t2	1:10	1:45	0:38
t3	1:35	1:20	0:46
<b>Średni czas</b>	<b>1:22</b>	<b>1:4</b>	<b>0:42</b>

Uśrednione wyniki ankiet otrzymane od uczestników badania przedstawione zostały w tab. 4.

Do wybranych kryteriów zostały przydzielone punkty zgodnie z uwzględnionymi kryteriami zawartymi w tab. 1 dla każdego z badanych narzędzi. System przydzielania punktów uwzględniał informacje zawarte na stronach internetowych producentów oprogramowania, jak również została przeprowadzona indywidualna weryfikacja każdego kryterium dla wybranych narzędzi. Wyniki zostały zaprezentowane w tab. 5.

Do ukazania jednoznacznych wyników analizy efektywnego prototypowania dla przedstawionych narzędzi przypisane zostały punkty do wybranych wcześniej parametrów. System naliczania punktów uwzględniał analizę każdego narzędzia pod kontem wcześniej ustalonych kryteriów z uwzględnieniem poszczególnych progów punktowania. Część punktów została uzyskana na podstawie uzyskanych ankiet po przeprowadzeniu normalizacji skali. tab. 6 przedstawia przydzielone punkty i znormalizowaną sumę wyników dla poszczególnego narzędzia.

Tabela 4. Wyniki ankiet

Treść pytania (skala)	Narzędzie do prototypowania		
	Pencil Project	Justinmind	Pidoco
1. Stopień trudności budowania prototypu (1-5)	1,67	2,33	2,67
2. Stabilność pracy narzędzia (1-5)	4	5	3,67
3. Szybkość działania programu (1-5)	4,67	4,67	4,33
4. Liczba dostępnych komponentów (1-4)	2,67	3,67	2,33

Tabela 5. Ocena narzędzi względem parametrów technicznych

Nazwa kryterium	Pencil Project (punkty)	Justinmind (punkty)	Pidoco (punkty)
Uruchomienie na systemach operacyjnych	1	1	1
Budowanie prototypów na wybrane platformy	1	1	1
Odzwierciedlenie interfejsu	0,2	0,5	0,3
Eksport prototypu	0,33	1	1
Cena narzędzia	0,7	0,3	0,3
<b>Suma wyników po normalizacji</b>	<b>0,7</b>	<b>0,82</b>	<b>0,79</b>

Tabela 6. Ocena narzędzi względem parametrów technicznych

Nazwa kryterium	Pencil Project (punkty)	Justinmind (punkty)	Pidoco (punkty)
Czas budowy prototypu	0,2	0,05	0,75
Wsparcie (poradniki, dokumentacja)	0,15	0,75	0,75
Testowanie na urządzeniach mobilnych	0	1	1
Interaktywność prototypu	0,10	0,75	0,75
Stopień trudności budowania prototypu	0,75	0,15	0,15
Stabilność pracy	0,75	0,75	0,15
Szybkość działania programu	0,75	0,75	0,75
Liczba dostępnych komponentów	0,15	0,75	0,15
<b>Suma wyników po normalizacji</b>	<b>0,35</b>	<b>0,51</b>	<b>0,59</b>

## 6. Analiza porównawcza narzędzi do prototypowania interfejsów

Na podstawie formuły (1) i (2) wyznaczone zostały wyniki dla parametrów technicznych i czynników efektywności. Oba główne parametry objęte analizą potraktowane zostały w sposób równoważny i przypisana została im jednakowa waga. Suma wyników po normalizacji przedstawia końcową liczbę uzyskanych punktów przez każde z narzędzi. Końcowe zestawienie punktów uzyskanych przez poszczególne narzędzie przedstawia tab. 7.

Tabela 7. Końcowe zestawienie wyników analizy porównawczej

Nazwa kryterium	Pencil Project (punkty)	Justinmind (punkty)	Pidoco (punkty)
Parametry techniczne	0,7	0,82	0,79
Parametry efektywności	0,35	0,51	0,59
Suma wyników po normalizacji	<b>0,53</b>	<b>0,67</b>	<b>0,70</b>

Analizując tab. 3 z uzyskanymi czasami można zauważyć, że każde badane narzędzie uzyskało inny czas, co przekłada się na różną efektywność budowania prototypu. Najniższy średni czas uzyskało narzędzie Pidoco. Na podstawie przeprowadzonej oceny poszczególnych narzędzi pod względem efektywności i po zsumowaniu wyników, można jednoznacznie stwierdzić, odczytując z tab. 7, że narzędzie Pidoco uzyskało najwyższą liczbę punktów. Głównym czynnikiem, który zdecydował o wyniku był uzyskany czas budowania prototypu. Drugim narzędziem, które uzyskało mniejszą liczbę punktów jest Justinmind.

Z przedstawionych wyników cząstkowych wywnioskować można, że narzędzie to cechuje rozbudowany interfejs, który posiada bardzo dużą liczbę komponentów w przeciwieństwie do pozostałych narzędzi. Narzędzie, które w tej klasyfikacji otrzymało najmniejszą liczbę punktów to oprogramowanie Pencil Projekt. Do najsłabszych cech można zaliczyć liczbę dostępnych komponentów czy interaktywność prototypu. Brak możliwości przetestowania prototypu na urządzeniach mobilnych również wpłynęła na niską ocenę.

## 7. Wnioski

Na podstawie przeprowadzonych badań i uzyskanych wyników zamieszczonych w ramach niniejszej pracy można stwierdzić, że narzędzia do prototypowania interfejsów cechują się różną efektywnością. Odwołując się do hipotezy, która mówi, że różne narzędzia do budowy prototypów interfejsów mają zróżnicowaną efektywność i dokonując analizy tab. 3 można jednoznacznie zauważyć, że każde prezentowane narzędzie uzyskało inny wynik. Oznacza to, że hipoteza H1 została udowodniona.

Dokonując analizy sformułowanych parametrów zamieszczonych w tab. 6, które realnie wpływają na efektywność prototypowania można odnieść się do drugiej hipotezy, która mówi, że możliwe jest sformułowanie

jednoznacznego kryterium do porównania efektywności narzędzia do budowy prototypów. W tym przypadku również druga hipoteza (H2) została udowodniona.

Przeprowadzając ostateczne porównanie prezentowanych narzędzi zamieszczonych w tab. 7 i odwołując się do postawionej tezy, która mówi, że istnieje możliwość wyboru optymalnego narzędzia do budowy prototypu interfejsu, odczytać można, że najbardziej optymalnym narzędziem dla przedstawionego problemu okazało się oprogramowanie Pidoco. Z uzyskanej analizy i wyników można jednoznacznie stwierdzić, że postawiona teza została potwierdzona.

Biorąc pod uwagę przeprowadzoną analizę, można wysnuć stwierdzenie, że wybór odpowiedniego narzędzia zależy od rodzaju projektu i wymagań stawianych, co do prototypowania. Każdy nowo budowany system wymaga indywidualnego podejścia w stosunku do procesu prototypowania. Stawiając odpowiednie warunki, możliwe jest wybranie odpowiedniego narzędzia, które jest w stanie sprostać oczekiwaniom projektu.

## Literatura

- [1] A. Sears, J. Jacko, Human-Comuter Interaction: Development Process, Taylor & Francis Group, 2009.
- [2] A. Leonidis, M. Antona, C. Stephanidis, Rapid Prototyping of Adaptable User Interfaces, International Journal of Human-Computer Interaction, 2012, 213-235.
- [3] K. Sacha, Inżynieria oprogramowania. Wydawnictwo naukowe PWN, 2010.
- [4] EvolutionaryPrototyping  
[http://www.teachict.com/as\\_a2\\_ict\\_new/ocr/A2\\_G063/331\\_systems\\_cycle/prototyping\\_RAD/miniweb/pg2.htm](http://www.teachict.com/as_a2_ict_new/ocr/A2_G063/331_systems_cycle/prototyping_RAD/miniweb/pg2.htm)  
(dostęp 05.2016r)
- [5] P. Weichbroth, M. Sikorski, User interface prototyping, techniques, methods and tools, Uniwersytet Ekonomiczny w Katowicach, 2015.
- [6] Justinmind: Prototyping platform for web and mobile apps  
<https://www.justinmind.com> (dostęp 07.2016)
- [7] Wireframing Design Tools- Pidoco  
<https://pidoco.com/en/tour/features> (dostęp 07.2016)
- [8] Prototype Tools- Pencil Project  
<http://pencil.evolus.vn/Features.html> (dostęp 07.2016)
- [9] T. Panek, Statystyczne metody wielowymiarowej analizy porównawczej. Szkoła Główna Handlowa - Oficyna Wydawnicza, 2009.
- [10] A. Sears, J. Jacko, Human-Comuter Interaction: Development Process. Taylor & Francis Group, 2009.
- [11] J. Arnowitz, M. Arent, N. Berger, Effective Prototyping For Software Makers. Amsterdam, Morgan Kaufmann, 2007.
- [12] A. Cid-López, M. Hornos, R. Carrasco, E. Herrera-Viedma, Applying a linguistic multi-criteria decision-making model to the analysis of ICT suppliers' offers. Expert Systems with Applications, 57, 2016, 127-138.
- [13] W. Bleek, M. Jeenicke, R. Klischewski, Developing Web-based Applications through e-Prototyping. Hamburg University, Department for Informatics, Software Engineering Group, 2002
- [14] L. Teixeira, V. Saavedra, C. Ferreira, J. Simoes, B. Santos, Requirements Engineering Using Mockups and Prototyping Tools: Developing a Healthcare Web-Application, Heidelberg, 2014, 652-663.