



PROGRAMOWANIE APLIKACJI MOBILNYCH

WYKŁAD 1

Wprowadzenie do aplikacji mobilnych

dr inż. Mateusz Pomianek

Katedra Informatyki i Automatyki, Politechnika Rzeszowska

m.pomianek@prz.edu.pl // [D.103](#) // <http://www.kia.prz.edu.pl> // <https://materialy.kia.prz.edu.pl>



1. Informacje o przedmiocie
2. Zaliczenie przedmiotu
3. Historia i ewolucja aplikacji mobilnych
4. Modele tworzenia: natywne, hybrydowe, cross-platform
5. Kluczowe narzędzia i technologie



***"If you want to make a pie from scratch,
you must first create the universe"***

– Carl Sagan



Informatyka

AA - Inżynieria systemów informatycznych

Technika informacyjno-pomiarowa

Pomiar a informacja, Błąd i niepewność pomiaru

Architektura systemów komputerowych - W30 L30

Architektura systemu komputerowego

Programowanie w języku C

C i C++



Wymagania formalne

- **Student zapisany na szósty semestr kierunku Informatyka studiów pierwszego stopnia**

Wymagania wstępne w kategorii Wiedzy

- **Znajomość obiektowych języków programowania np. C++, C#, Java, Swift, Kotlin.**

Wymagania wstępne w kategorii Umiejętności

- **Obsługa komputera PC i iMac**

Wymagania wstępne w kategorii Kompetencji społecznych

- **Uczciwość, umiejętność pracy w zespole, świadomość funkcjonowania społeczeństwa**



Wykłady - 25h

1. Wprowadzenie do aplikacji mobilnych
2. Projektowanie UI/UX dla mobile
3. Architektura aplikacji mobilnych
4. Android: programowanie natywne
5. iOS: programowanie natywne
6. Flutter i Dart: cross-platform
7. React Native: cross-platform
8. Monitorowanie wydajności i testowanie aplikacji mobilnych
9. Integracja z backendem
10. Publikacja i marketing aplikacji

Laboratoria - 15h

1. Idea i pierwszy szkic – fundamenty aplikacji
2. Projektowanie UI/UX oraz architektury
3. Pierwsze funkcje: formularze, listy, edycja i usuwanie danych
4. Komunikacja z serwerem poprzez REST API
5. Funkcje mobilne: lokalizacja, kamera czy powiadomienia
6. Testowanie i cross-platform
7. Publikacja, marketing i obrona projektu



Forma zajęć	Praca przed zajęciami	Udział w zajęciach	Praca po zajęciach
Wykład	<ul style="list-style-type: none">Przygotowanie do obrony 2 godz./sem	<ul style="list-style-type: none">Godziny 25.00 godz./sem	<ul style="list-style-type: none">Uzupełnienie/studiowanie notatek 1 godz./sem.Studiowanie zalecanej literatury 10 godz./sem.Inne: 5 godz./sem.
Laboratorium	<ul style="list-style-type: none">Przygotowanie do laboratorium 7 godz./sem.Przygotowanie do obrony 5 godz./sem.	<ul style="list-style-type: none">Godziny 15 godz./sem.	<ul style="list-style-type: none">Inne: 15 godz./sem.
Zaliczenie		<ul style="list-style-type: none">Zaliczenie 1.00 godz./sem.	



	Poniedziałek	Wtorek	Środa
7:00			
8:00			
9:00			8:45, gr.1 Programowanie aplikacji mobilnych - Wykład (s.B.107 bud.B)
10:00			
11:00			
12:00			
13:00			
14:00		14:00, gr.3 /raz na dwa tygodnie - nieparzyste/ Programowanie aplikacji mobilnych - Laboratorium (s.F.105 bud.F)	14:00, gr.2 /raz na dwa tygodnie - parzyste/ Programowanie aplikacji mobilnych - Laboratorium (s.F.105 bud.F)
15:00			
16:00		15:45, gr.1 /raz na dwa tygodnie - parzyste/ Programowanie aplikacji mobilnych - Laboratorium (s.F.105 bud.F)	
17:00			

1. Wykład, 25 godzin

○ -> 12 wykładów + obrona projektu

2. Laboratorium, 15 godzin

○ -> 7 ćwiczeń lab. + obrona projektu

Sposób wystawiania oceny	Wymagania konieczne do zaliczenia przedmiotu
<p data-bbox="61 379 563 486">Wykład</p> <p data-bbox="137 525 996 682">Przygotowanie i obrona dokumentacji projektowej dla własnej aplikacji mobilnej.</p>	<ol data-bbox="1034 396 2308 675" style="list-style-type: none">1. Brief aplikacji (<i>1 kartka A4</i>)2. Zespół wymagań biznesowych, BRD (<i>1 kartka A4</i>)3. Zespół wymagań funkcjonalnych, FRD (<i>1 kartka A4</i>)4. Zespół wymagań technologicznych, TRD (<i>1 kartka A4</i>)
<p data-bbox="61 768 563 875">Laboratorium</p> <p data-bbox="137 943 996 1200">Uczestnictwo w zajęciach. Na przedostatnich i ostatnich zajęciach prezentowany jest projekt demonstrujący umiejętności.</p>	<ol data-bbox="1034 746 2257 1343" style="list-style-type: none">1. Aplikacja utworzona w jednym z języków:<ul data-bbox="1085 803 2257 846" style="list-style-type: none">• C++, Unity C#, Java, Kotlin, Swift, React Native, Flutter,2. Interfejs użytkownika<ul data-bbox="1085 918 1875 1061" style="list-style-type: none">• Nawigacja, kilka ekranów, zakładki,• Przyciski, ikony, grafiki,• Formularze, listy,3. Obsługa danych wewnętrznych<ul data-bbox="1085 1132 2155 1232" style="list-style-type: none">• Odczyt, edycja, usuwanie i zapis danych z plików,• Obsługa danych z sensorów IMU, GPS, kamery,4. Obsługa danych zewnętrznych<ul data-bbox="1085 1303 2028 1343" style="list-style-type: none">• Komunikacja i pobieranie danych z serwera,



	WYKŁADY	LABORATORIA
Ocena 3	<p>Dokumentacja projektowa (4xA4) +</p> <ul style="list-style-type: none"> • Indywidualna obrona 	<p>Projekt aplikacji wraz z kodem programu +</p> <ul style="list-style-type: none"> • Indywidualna obrona, • Aplikacja posiada wymagane funkcjonalności,
Ocena 4	<p>Dokumentacja projektowa (4xA4) +</p> <ul style="list-style-type: none"> • Indywidualna obrona • Racjonalne uzasadnienie podjętych wyborów i zastosowanych rozwiązań 	<p>Projekt aplikacji wraz z kodem programu +</p> <ul style="list-style-type: none"> • Indywidualna obrona, • Aplikacja posiada wymagane funkcjonalności, • Przemyślana architektura i testy,
Ocena 5	<p>Dokumentacja projektowa (4xA4) +</p> <ul style="list-style-type: none"> • Publiczna obrona (prezentacja 5 min) • Racjonalne uzasadnienie podjętych wyborów i zastosowanych rozwiązań 	<p>Projekt aplikacji wraz z kodem programu +</p> <ul style="list-style-type: none"> • Indywidualna obrona, prezentacja działającej aplikacji na urzędzeniu, • Aplikacja posiada wymagane oraz dodatkowe funkcjonalności, • Przemyślana architektura i testy,
Ocena końcowa	Średnia 50/50 (gdy oba moduły są zaliczone przynajmniej na 3)	



Wykład	folder	☰ węzeł	☰ dodaj element	☰ dane
Projekt aplikacji mobilnej	folder	☰ węzeł	☰ dodaj element	☰ dane
Brief aplikacji	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Zespół wymagań biznesowych (BRD)	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Zespół wymagań funkcjonalnych (FRD)	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Zespół wymagań technologicznych (TRD)	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Publiczna prezentacja projektu aplikacji (na ocenę 5)	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →

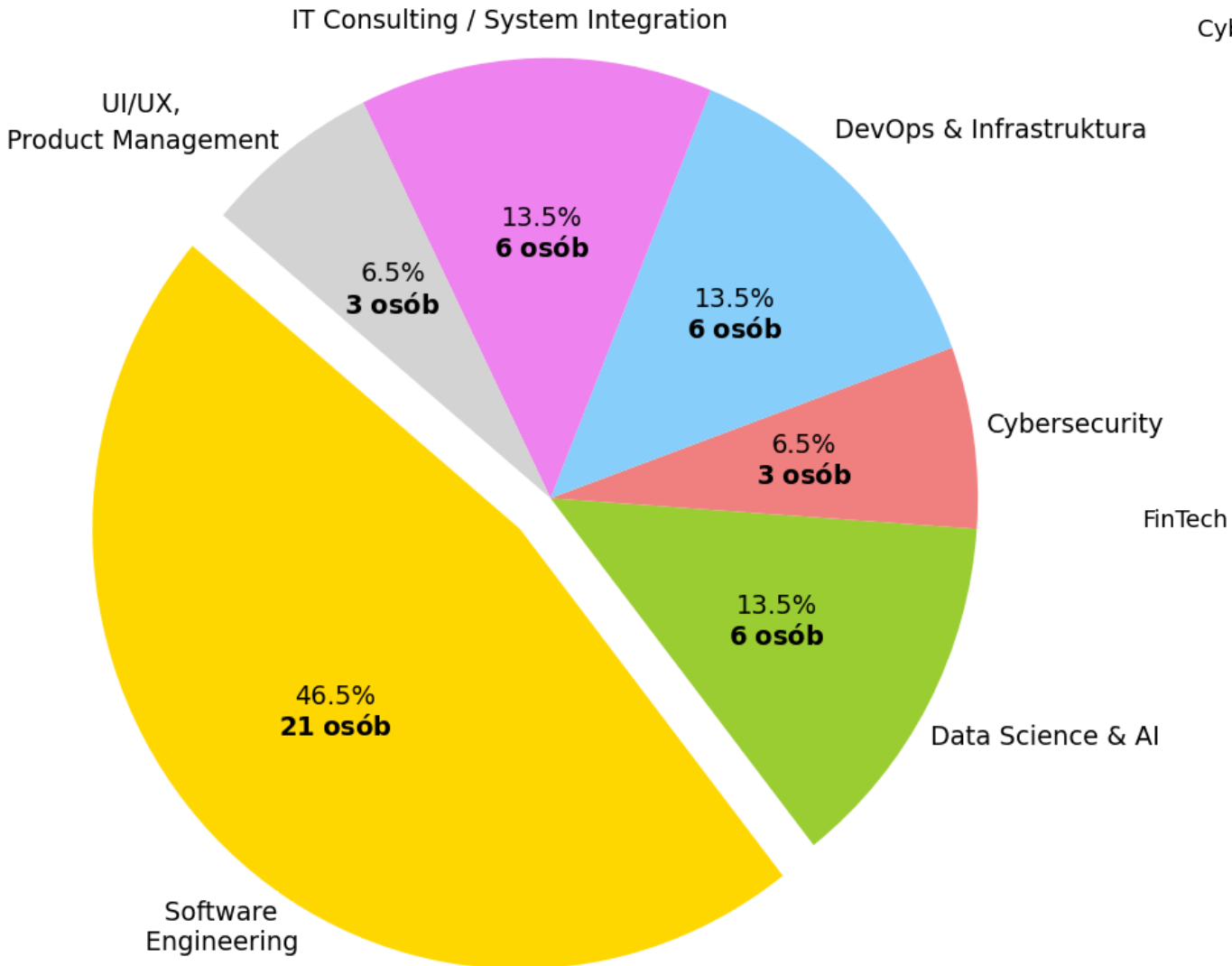
Laboratorium	folder	☰ węzeł	☰ dodaj element	☰ dane
Obrona własnej aplikacji mobilnej	folder	☰ węzeł	☰ dodaj element	☰ dane
Nawigacja pomiędzy ekranami i Interfejs użytkownika (UI)	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Architektura aplikacji	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Multimedia i animacje	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Sensory i kamera	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →
Korzystanie z zewnętrznych bibliotek	10.0 pkt	☰ węzeł	☰ dane	wyniki zadania →

Punkty	Ocena
0 - 50	2
51 - 60	3,0
61 - 70	3,5
71 - 80	4,0
81 - 90	4,5
91 - 100	5,0

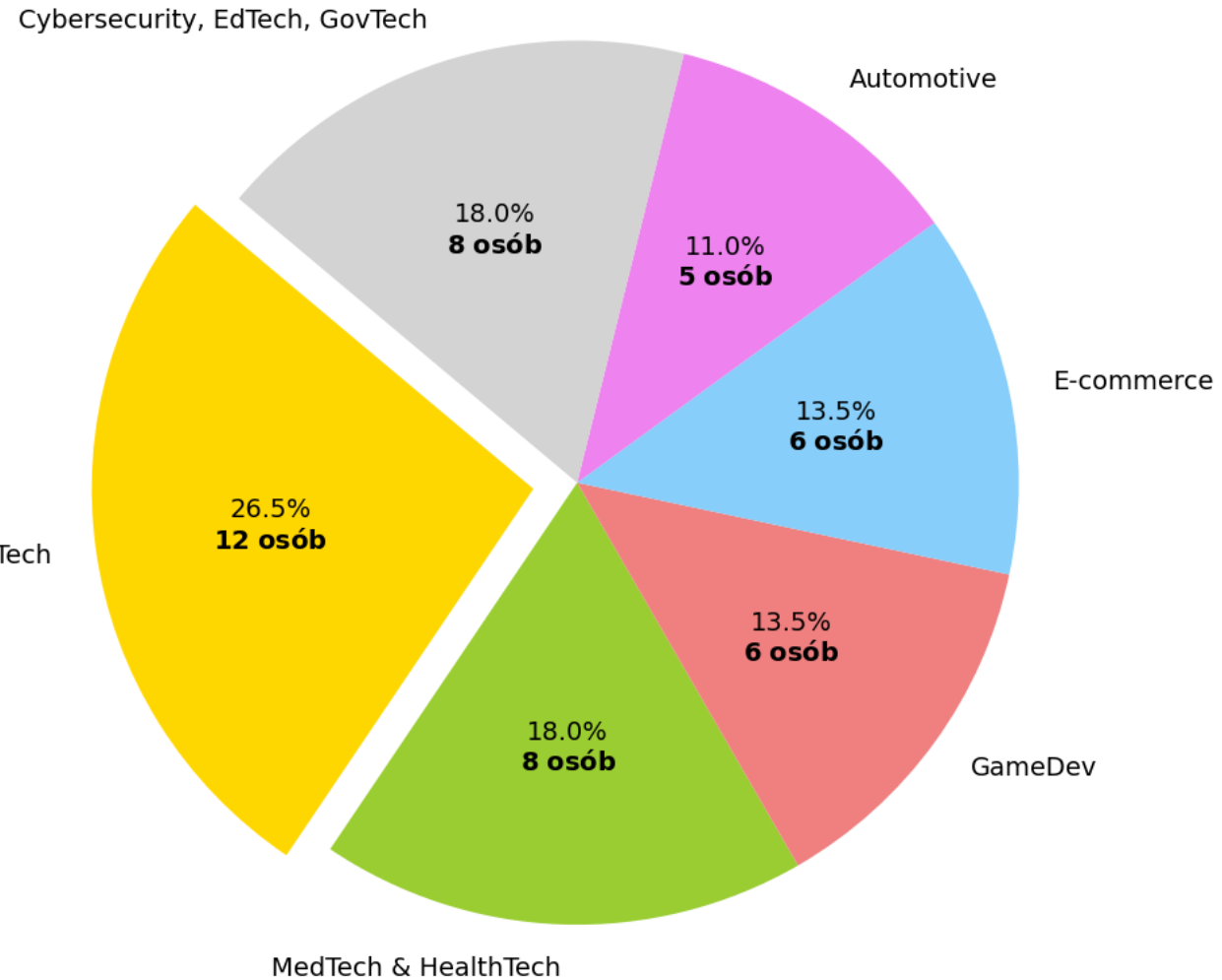
Wykład	Laboratoria	Ocena końcowa
0 - 50 pkt	0 - 50 pkt	0 - 100 pkt



**Rodzaj pracy absolwentów Informatyki
(45 osób)**

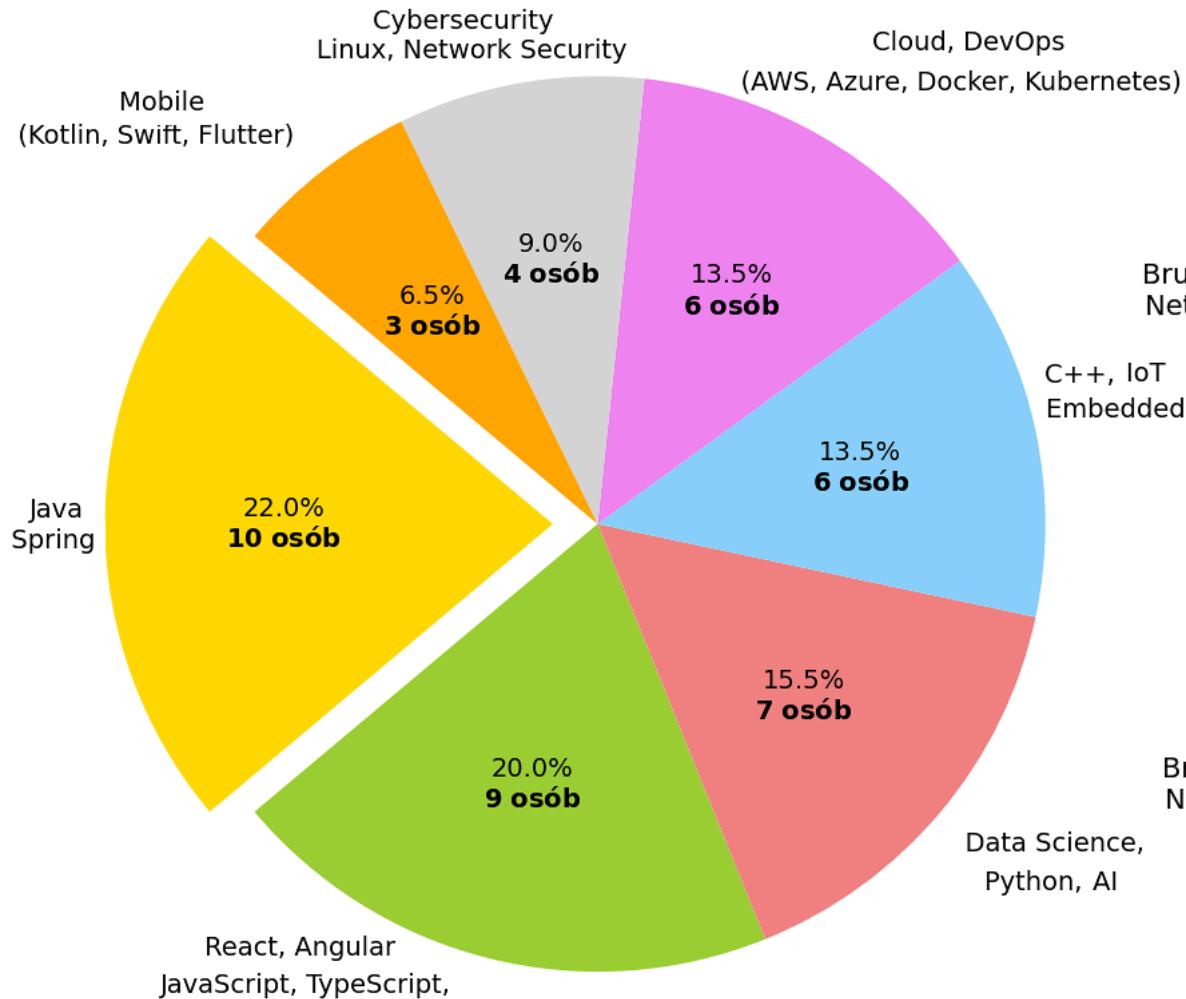


**Branże wybierane przez absolwentów Informatyki
(45 osób)**

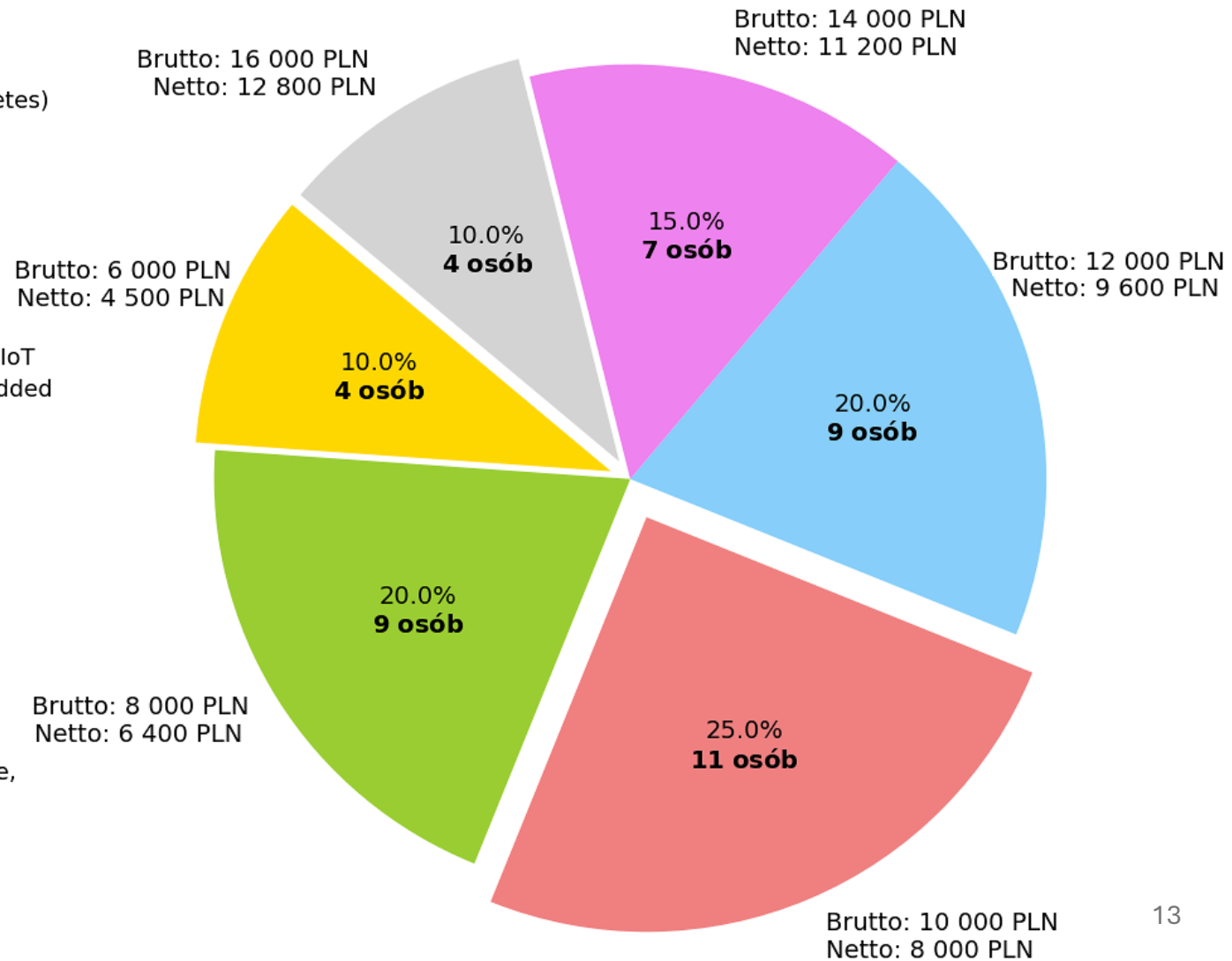




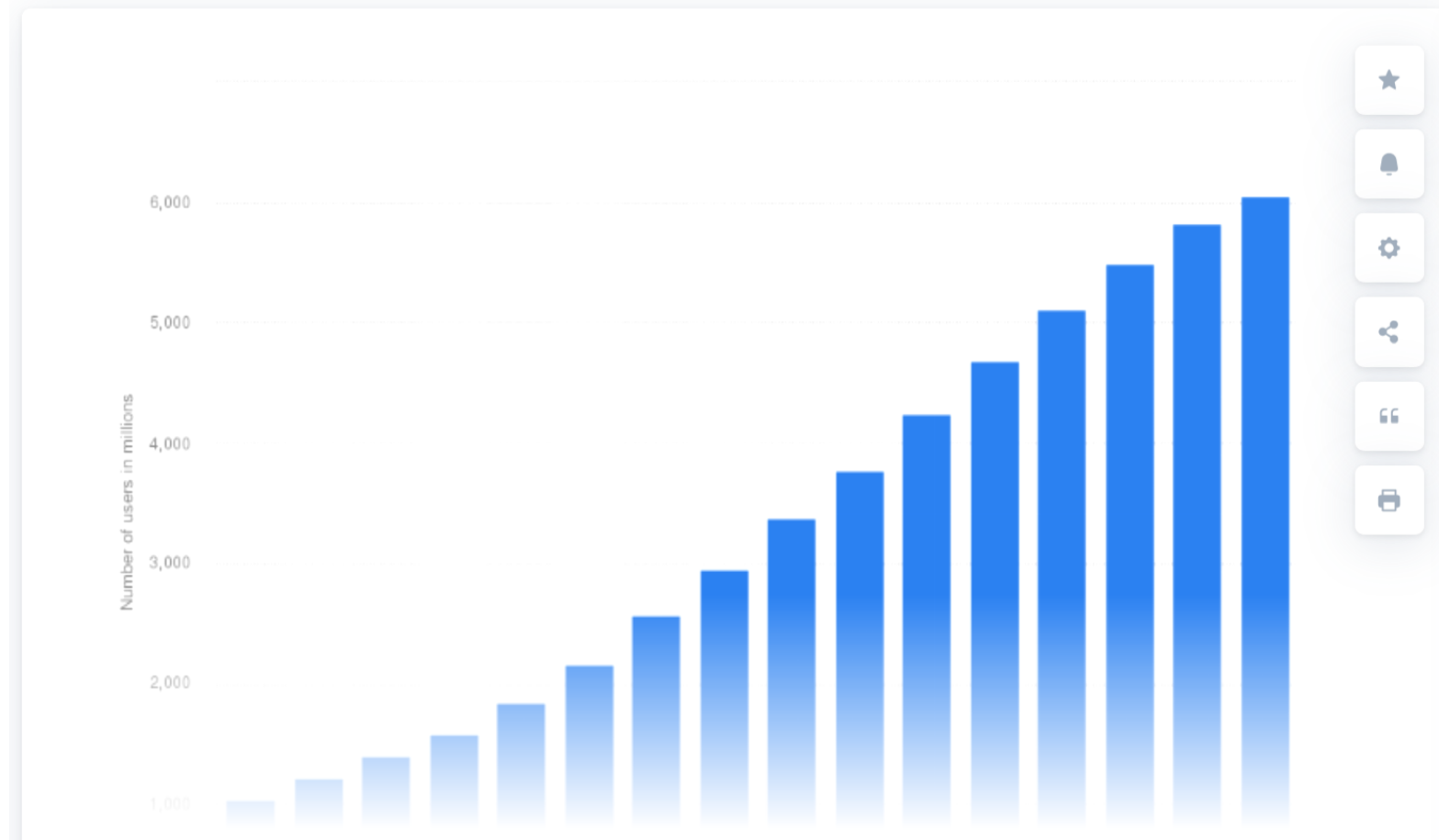
**Technologie używane przez absolwentów Informatyki
(45 osób)**



**Zarobki absolwentów Informatyki w Polsce (5-10 lat doświadczenia)
(45 osób)**

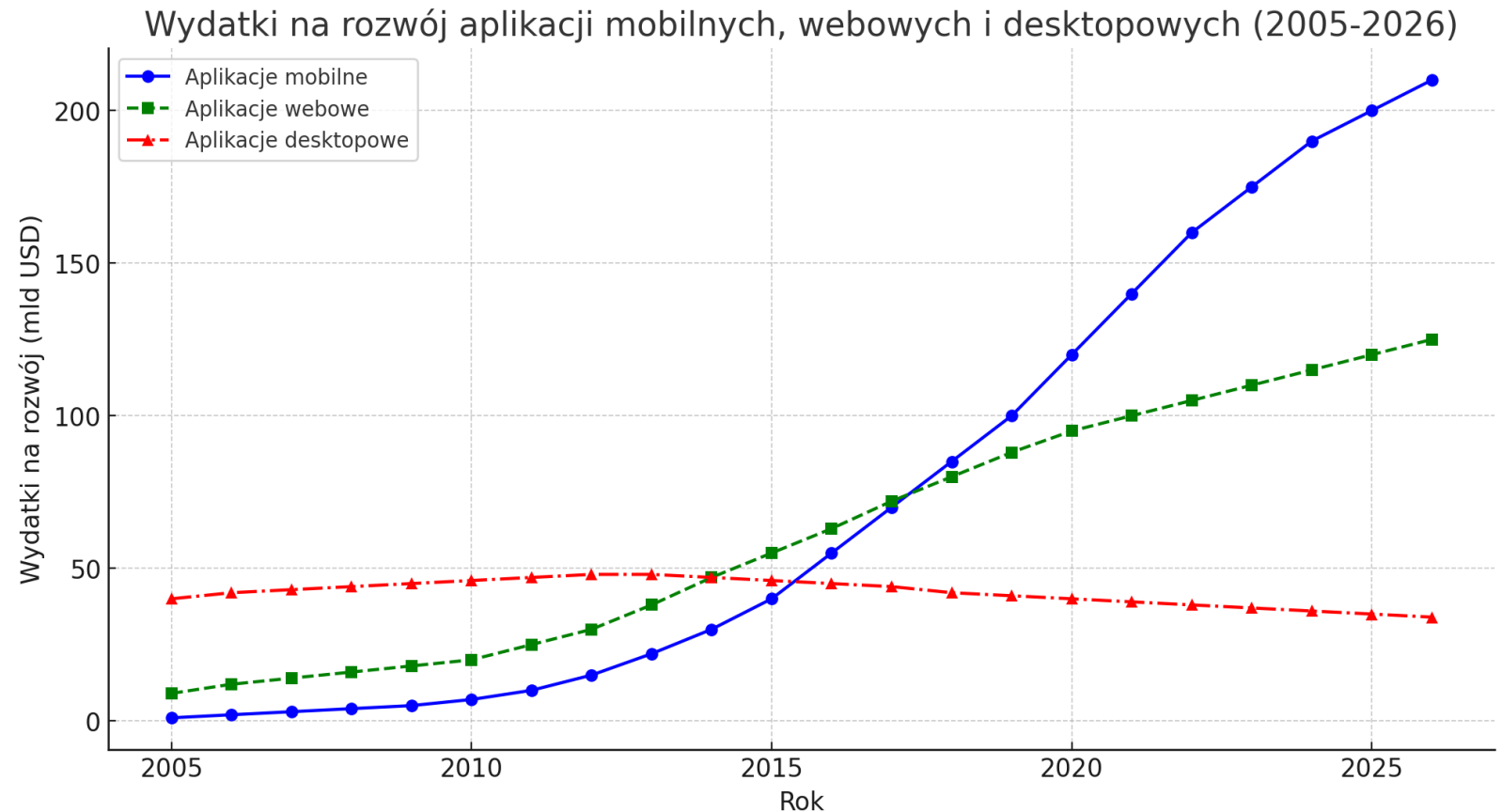


Number of smartphone users worldwide from 2014 to 2029 *(in millions)*



Trend ogólny:

- **2000-2010** Dominacja aplikacji desktopowych
- **2010-2020:** Ogromny wzrost wydatków na aplikacje mobilne i webowe
- **2020-2026:** Utrzymanie wysokiego poziomu wydatków na mobilne i webowe, ale desktopy schodzą na dalszy plan



Całkowite nakłady finansowe przeznaczone na tworzenie, utrzymanie i rozwój aplikacji w danym segmencie (mobilnym, webowym i desktopowym) w ujęciu globalnym. Obejmuje to:

Aplikacje desktopowe

- Koszty tworzenia oprogramowania na Windows, macOS i Linux
- Wydatki na licencjonowanie narzędzi (np. Microsoft, Adobe, Autodesk)
- Rozwój i aktualizacje istniejących programów (np. Office, Photoshop, AutoCAD)
- Nakłady na optymalizację i kompatybilność z nowymi systemami

Aplikacje webowe

- Nakłady na rozwój frontend (HTML, CSS, JS, React, Angular) i backend (Node.js, Django)
- Koszty infrastruktury (serwery, chmura – AWS, Azure, GCP)
- Wydatki na zabezpieczenia i optymalizację (SEO, CDN, SSL)
- Utrzymanie i rozwój Progressive Web Apps (PWA)

Aplikacje mobilne

- Koszty projektowania i programowania (iOS, Android, cross-platform)
- Inwestycje w technologie mobilne (np. frameworki jak Flutter, React Native)
- Wydatki na testowanie i optymalizację (np. testy UX, kompatybilność urządzeń)
- Koszty utrzymania, aktualizacji i wsparcia technicznego



- **2010-2015:** W tym okresie nastąpił dynamiczny wzrost popularności smartfonów i tabletów, co skłoniło firmy do inwestowania w rozwój aplikacji mobilnych. Wydatki na aplikacje webowe i desktopowe rosły wolniej, ponieważ uwaga deweloperów i użytkowników skupiała się na mobilności.
- **2016-2020:** Aplikacje mobilne dominowały na rynku, a firmy przeznaczają znaczne budżety na ich rozwój. W 2020 roku średni koszt stworzenia aplikacji mobilnej w Ameryce Północnej przekraczał 300 000 USD.
- itcraftapps.com
- Wydatki na aplikacje webowe również rosły, zwłaszcza dzięki rozwojowi technologii takich jak Progressive Web Apps (PWA), które łączyły zalety aplikacji webowych i mobilnych. Inwestycje w aplikacje desktopowe były mniejsze, ale utrzymywały się na stabilnym poziomie.



- **Kotlin:** Ten nowoczesny język programowania jest naszym pierwszym wyborem, gdy tworzymy natywne aplikacje dla systemu Android. Kotlin pozwala nam na szybkie i efektywne tworzenie aplikacji, które są niezawodne, wydajne i przyjazne dla użytkownika.
- **Swift:** Kiedy tworzymy aplikacje na urządzenia Apple, korzystamy ze Swifta. Dzięki temu językowi możemy tworzyć szybkie, wydajne i bezpieczne aplikacje, które spełniają najwyższe standardy Apple.
- **React Native:** Jeżeli chcesz jednocześnie aplikacji na Androida i iOS, wybieramy React Native. Dzięki temu frameworkowi możemy stworzyć jednolitą, spójną i wydajną aplikację na obie platformy, co pozwala zaoszczędzić czas i zasoby.
- **PWA (Progressive Web Apps):** To technologia, która pozwala na stworzenie aplikacji internetowej, która zachowuje się jak natywna aplikacja mobilna. PWA to świetna opcja, jeżeli chcesz zapewnić użytkownikom doświadczenia zbliżone do natywnych aplikacji, niezależnie od urządzenia, na którym korzystają z aplikacji.
- **Flutter**
- **C#**



"Brief" aplikacji mobilnej - wytyczne projektu

1. Czym ma być aplikacja mobilna, co ma robić, jaki jest jej cel
2. Kim jest użytkownik końcowy, jaki jest kontekst użycia
3. Charakter i główne funkcje aplikacji
4. Treść aplikacji
5. User experience, UX (*doświadczenie użytkownika*)
6. Wygląd aplikacji
7. Szczegóły techniczne
8. Budżet i ramy czasowe realizacji



Żeby rozpocząć projekt, potrzebne są biznesowe, funkcjonalne i technologiczne wymagania.

1. Zespół wymagań biznesowych (BRD)
2. Zespół wymagań funkcjonalnych (FRD)
3. Zespół wymagań technologicznych (TRD)

Chociaż on stanowi tak naprawdę w cenie wytworzenia aplikacji, czy czasochłonności, można powiedzieć, że około 10%, to jest prototypowanie, około 70% kosztu to jest koszt pracy developerów, 10% to koszt zarządzania projektem komunikacji i 10% to testowanie. Faktycznie te pierwsze 10% jest o tyle ważne, że jeżeli nawalimy tutaj, schrzanimy robotę i coś źle zaprototypujemy, to potem coraz trudniej to odkręcić. To znaczy koszt odkręcenia na etapie testów, w porównaniu z kosztem odkręcenia na etapie prototypowania są dramatycznie różne.



Czyli jak stworzyć program i nie stracić

Minimalnie opłacalny produkt MVP to wersja produktu posiadająca wystarczającą liczbę funkcji, aby nadawała się do użytku przez pierwszych klientów, którzy mogą następnie przekazać informacje zwrotne na temat przyszłego rozwoju produktu.



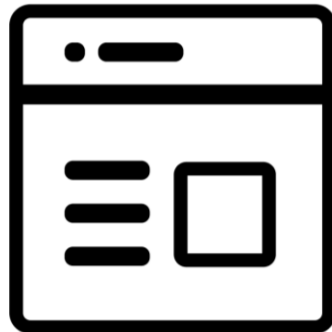
- 1. Rozpoznanie potrzeb:** Na początek chcemy zrozumieć wizję, cele biznesowe i oczekiwania. Możemy wtedy stworzyć strategię, która najlepiej spełni wymagania.
- 2. Warsztaty projektowe:** To etap, na którym dokładnie analizujemy problem, który ma rozwiązać aplikacja. Wspólnie definiujemy grupę docelową, określamy kluczowe funkcjonalności i zaczynamy pracę nad strukturą aplikacji.
- 3. Projektowanie UX/UI:** Stworzenie intuicyjnego interfejsu użytkownika
- 4. Programowanie:** Na tym etapie programiści przekształcają projekt w działający produkt. Korzystamy z najnowszych technologii i najlepszych praktyk programistycznych, aby zapewnić wysoką jakość i efektywność aplikacji.
- 5. Testy:** Testy na różnych urządzeniach i systemach operacyjnych.



Projektowanie **interfejsu użytkownika (UI)** odnosi się do procesu tworzenia estetycznego, funkcjonalnego i intuicyjnego **układu elementów graficznych**, takich jak **przyciski, pola tekstowe, ikony** i inne **elementy interaktywne**, które **umożliwiają użytkownikom komunikację z systemem**.

Głównym celem który obiera projektant UI w codziennej pracy jest stworzenie interfejsu, który jest łatwy oraz przyjazny w obsłudze i dostarcza satysfakcjonującego doświadczenia wedle potrzeb klienta.

Splash screen (ekran powitalny) – obrazek wyświetlany na ekranie lub jego części podczas uruchamiania bądź ładowania danej aplikacji.



Wireframe



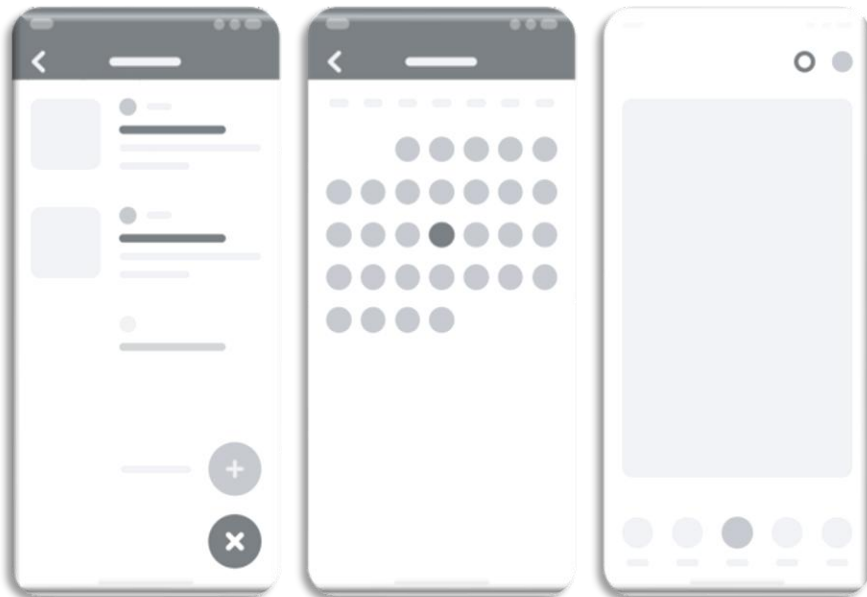
Makieta



Prototyp

UX Design definiuje kształt interfejsu użytkownika (UI Designu)

Głównym celem UX Designu jest zapewnienie użytkownikom intuicyjnego i satysfakcjonującego doświadczenia podczas korzystania z produktu lub usługi. W procesie tworzenia interfejsu użytkownika, UX Designer bada i analizuje zachowania, potrzeby i oczekiwania użytkowników, aby lepiej zrozumieć, jakie elementy interfejsu są dla nich najważniejsze.



Na podstawie tych wniosków, UX Designer projektuje strukturę informacji, przepływ użytkownika, a także funkcjonalności i interakcje, które poprawią użyteczność i użyteczność interfejsu.



Rodzaj aplikacji może znacząco wpłynąć na wybór strategii monetyzacji.

Na przykład:

- Aplikacje typu „freemium” mogą być idealnym rozwiązaniem dla gier, gdzie użytkownicy mogą być zainteresowani odblokowywaniem dodatkowych funkcji lub poziomów.
- Dla aplikacji biznesowych model subskrypcji może okazać się bardziej trafionym wyborem.



Z posiadaniem aplikacji mobilnej jest jak z posiadaniem dobrego samochodu, to znaczy, trzeba na to wydać pieniądze. Zresztą ta metafora jest bardzo stosowna też, jeżeli chodzi o wydatki. Dlatego, że **aplikacja mobilna** i jej stworzenie, **kosztuje** mniej więcej **tyle, co zakup samochodu**.

1. Można zrobić aplikację za **50 tysięcy złotych**, czyli za tyle, za ile kupimy *Fiata Tipo* z klimatyzacją i radiem, ale nie będziemy mieli tam jakiś super bajerów.
2. Jeśli chcemy zrobić bardziej zaawansowaną aplikację, wydamy na to **100, 200, 300 tysięcy**.
3. A jeżeli oczywiście chcemy mieć *Maybacha*, bo to ma być potężna aplikacja, rozbudowana, łączyć się z wieloma systemami, to tak jak *Maybach*, będziemy musieli za nią zapłacić **milion złotych**.