

# Kategorie analizy frameworków GUI

<b>Ogólne</b>	<b>3</b>
<b>Komponenty</b>	<b>3</b>
<b>IT</b>	<b>4</b>
<b>Analiza pod kątem</b>	<b>4</b>

Ten dokument stanowi punkt wyjścia do analizy frameworków GUI. Wynikiem analizy będzie wybór systemu, który obsłuży przyszłą bibliotekę.

## Ogólne:

- Nacisk komunikacyjny (jak komunikuje się światu każdy z frameworków, co uważa za swoją najmocniejszą stronę etc.)
- Główne zastosowanie (w czym dany framework jest najsilniejszy)
- Kontekst mobilny (framework powinien to zapewniać)
- Open source (to warunek kluczowy do poszerzania biblioteki)
- Popularność (aktywna społeczność pomaga rozwiązać wiele problemów)
  - rozwijany
  - jest społeczność
  - pytania i odpowiedzi na stacku
- WCAG 2.0 / dostępność - deklaracja (to warunek administracji państwowej)
- Testy (w jaki sposób były prowadzone, w jakim zakresie)
- Poziom trudności / dla kogo (należy pamiętać o przyszłych dostawcach)
- Pros and cons (jeśli uda się zebrać takie informacje z perspektywy używania danego fw)
- Kompatybilność (administracja publiczna nie dysponuje najnowocześniejszym sprzętem)
- Dodatki (jak dużo jest dostępnych gotowych dodatków)

## Komponenty

- Dostępne layouty i gridy
- Modele nawigacji
- Komponenty UI
  - Headery
  - Bloki tekstu
  - Inputy
  - Kontrolki (radiobuttony, checkboxy etc.)
  - Feeds
  - Discussion feeds
  - Tabele
  - Media
  - Toolbars
  - Widgets
- Typografia
- Kolory
- Mikrointerakcje
- System notyfikowania użytkownika
- Ikony
- Onboarding

## IT

*Kryteria oceny, na podstawie których programista będzie oceniał frameworki frontendowe. Celem analizy jest decyzja, czy piszemy od nowa framework, czy korzystamy z gotowego np. Salesforce.*

Jako efekt analizy powinna się też pojawić szacowana "Pracochłonność stworzenia od nowa" - będzie ona podstawą do decyzji o wykorzystaniu gotowego rozwiązania lub tworzeniu własnego.

Należy założyć, że wymogi minimalne są znacznie poniżej standardu rynkowego. Nie działa w tym wypadku zasada statystyczna, są jednostkowe przypadki, które determinują kompatybilność ze starym sprzętem (IE 8/9, 0.5GB RAM ...). Raczej skłaniamy się do języka sass.

Do określenia:

- Ile własnego css'a będziemy potrzebować?
- Statystyczne wykorzystanie frameworków w sieci - zbadanie fundamentu popularności danego fw. np.  
<https://trends.google.pl/trends/explore?cat=1142&q=%2Fm%2F0j671ln.Foundation.Material%20UI.Semantic-UI.Materialize>  
<https://trends.google.pl/trends/explore?cat=1142&q=Material%20Design.Lightning%20Design%20System.%2Fm%2F0j671ln>
- Wykorzystanie systemu gridowania uznanego za standard (np. Susy) pozwoli na uniknięcie nieporozumień w wykorzystaniu fw GUI.
- Wykorzystanie kilku frameworków jednocześnie - możliwa opcja

## Analiza pod kątem

- Dokumentacja (Czy jest i czy jest dobra, tutoriale/przykłady)
- Działanie zgodne z dokumentacją
  - Brak niezaimplementowanych rzeczy
  - Brak błędów
  - Brak luk bezpieczeństwa
- Dobra architektura
  - Separacja komponentów
  - Dobre, systematyczne, intuicyjne nazewnictwo komponentów, klas, ...
  - Brak niedobrych rozwiązań/antywzorców
- Możliwość integracji z innymi technologiami
  - różnymi frameworkami JS (nawet tymi, które jeszcze nie powstały)
  - Transpilerami less, sass (tzn. style można napisać w tych językach, skompilować i użyć)
- Łatwość budowania projektów na podstawie frameworku
  - eksperyment programistyczny

- Czy można framework skutecznie (efekt jest dobry, używalny równie łatwo, jak oryginał), łatwo (bez “hacków”) modyfikować/customizować.
  - Jakość kodu frameworku nie może przeszkadzać w modyfikacjach (tzn. musi być dobra).
  - Taka modyfikacja może być łatwo przeniesiona na nowszą wersję frameworku.
  - Czy jest wiele przykładów modyfikacji frameworku. Stworzenia innych komponentów na podstawie istniejących
  - Czy jest gotowa biblioteka komponentów?
  - Jak dużo będziemy mieli custom CSS i czy lepszy jest rozbudowany czy mały framework?
  - Czy są określone best practices, które pomogą utrzymać długookresową jakość kodu i wsparcie użytych rozwiązań?
- Wsparcie
  - Jaka jest najstarsza wersja przeglądarki, którą należy wspierać? Część przeglądarek nie wspiera pewnych elementów HTML5 i CSS3, najczęściej używana jest przeglądarka Internet Explorer - czy framework wspiera wymagane rozwiązania?
  - Czy ma sensowny cross-browser support, tj. czy elementy nie rozjeżdżają się zaledwie w różnych przeglądarkach?
- Język - SASS, LESS, pure CSS?