



Przekazywanie projektów graficznych do prac frontendowych

Zbiór porad i dobrych praktyk, które usprawniają współpracę
pomiędzy projektantami i programistami

Wstęp

Jako projektanci UI/UX wierzymy, że projekty które przygotowujemy są najlepszym rozwiązaniem problemów naszych klientów. Chcemy, aby efekty naszej pracy pokrywały się z wyglądem i działaniem finalnego projektu, a sam proces wdrożenia był maksymalnie przejrzysty i przyjemny dla wszystkich, którzy w nim uczestniczą.

Jak korzystać z tego dokumentu

Zdajemy sobie sprawę z tego, że tekstu jest sporo, a nasze projekty są tak różnorodne, że nie da się stworzyć dokumentu, który będzie dotyczył wszystkich idealnie.

Zacznij od przejrzania spisu treści, aby zlokalizować fragmenty które mogą dotyczyć Twojego projektu. Tam gdzie było to możliwe staraliśmy się podsumowywać paragrafy w pierwszym **wytłuszczonym** zdaniu, żeby ułatwić Ci przeglądanie i wyszukiwanie ważnych fragmentów.

Zgłaszanie błędów i sugerowanie pomysłów

Źródła tego dokumentu są dostępne w formacie **Markdown** pod adresem <https://gitlab.polcode.com/ui-ux/best-practices-frontend>. Zachęcamy do wysyłania PRóów z porawkami i pomysłami.

Przekazanie projektu

Spotkanie z developerem

Przekazanie projektu powinno rozpocząć się od spotkania (zwykle w formie telekonferencji z współdzielonym przez projektanta ekranem), podczas którego projektant zapoznaje frontendowca ze strukturą przekazywanego projektu i przechodzi przez poszczególne ekrany/flowy. Naturalnie ilość informacji przekazywana podczas jednego spotkania będzie ogromna – nie wymagamy zrozumienia/zapamiętania wszystkiego.

Omawianie projektu w ten sposób pozwala osobie odbierającej na zapoznanie się z pełnym zakresem projektu i jego niuansami (np. powody podjęcia pewnych decyzji podczas projektowania, dlaczego część ekranów odbiega układem od reszty).

Format

Przekazując pliki lub dostęp do projektów w chmurze upewnijmy się, że osoba, która je otrzymuje będzie je w stanie otworzyć. Dla obniżenia progu wejścia (tam gdzie to możliwe), zalecamy korzystanie z rozwiązań cloud-based, takich jak [Figma](#), [Adobe XD](#), [Sketch](#), czy [Zeplin](#) lub [Invision](#). Jeśli to niemożliwe, przekonwertujmy projekty do formatu, który odbierający będzie mógł otworzyć na swoich urządzeniach.

Struktura

Ekrany/artboardy powinny być uporządkowane. Decyzję o tym czy powinny być ułożone tematycznie (np. wszystkie ekrany związane z profilem użytkownika ustawione blisko siebie), czy logicznie (sekwencja ekranów odpowiadająca za flow płatności) czasami zależy od specyfiki projektu i pozostawiamy ją projektantowi.

Decyzję o tym czy ekrany responsywne (mobile, tablet, desktop) powinny być ułożone obok siebie, czy być pogrupowane rozmiarami na osobnych stronach pozostawiamy projektantowi i zespołowi – czasami ta sama osoba wdraża je wszystkie, a innym razem powstają odrębne aplikacje (np. web i React Native), które wdrażają różne osoby (i wygodniej jest im się obracać tylko w ramach swoich zadań). Najważniejsza jest tu wygoda zespołu i konsekwencja.

Dobrym zabiegiem jest dodanie dużych nagłówków (ok. 200pt) opisujących grupy ekranów – dzięki temu robiąc zoom out (aby zobaczyć wszystkie widoki) łatwiej odnajdujemy się w projekcie.

Pamiętajmy, aby ukończone ekrany dokładnie nazywać. Pomaga to w ich wyszukiwaniu i przeglądaniu. O ile w przypadku kilkunastu ekranów nawigowanie nawet po nieopisanych ekranach jest to łatwe, o tyle w przypadku projektów posiadających po kilkaset widoków konsekwencja w nazewnictwie zaprocentuje i ułatwi pracę wszystkim osobom zaangażowanym w proces tworzenia produktu.

Projektuj nowe widoki w miejscu z dala od już gotowych. W przypadku narzędzi pozwalających na kolaborację, staraj się wydzielić obszar zawierający już ukończone widoki (np. używając Pages w przypadku Figma i Sketcha). Pozwoli to uniknąć nieporozumień które z ekranów są już gotowe, a nad którymi jeszcze trwają prace.

Interaktywne elementy interfejsu i przejścia pomiędzy ekranami. Jeśli program graficzny w którym przygotowujemy materiały nam na to pozwala (np. Figma, Sketch), warto poświęcić czas na zamarkowanie interakcji poszczególnych elementów interfejsu (np. kliknięcie w przycisk powoduje przejście do kolejnego ekranu lub pojawienie się pop up'u). Dzięki temu będziemy w stanie:

- Samodzielnie odkryć brakujące ścieżki użytkowników
- Dostarczyć klientowi “prototyp” aplikacji, co pozwoli mi na zweryfikowanie poprawności pierwotnych założeń i ewentualną szybką zmianę kursu zanim projekt trafi do wdrażania
- Dostarczyć programistom dodatkowy poziom informacji (w postaci interaktywnej dokumentacji), co zdecydowanie ułatwi im wybór.

Weryfikacja jakości wdrożenia

Wspólna praca projektanta i programisty nie powinna się kończyć na przekazaniu projektu graficznego. W trakcie procesu wdrożenia powinno się odbyć conajmniej jedno spotkanie (choć zalecamy robienie ich cyklicznie), na którym grafik będzie mógł zweryfikować jakość wdrożenia i zgodność z przekazanym wcześniej projektem.

Dobłą praktyką są też cykliczne prezentacje wyglądu wdrażanego projektu product ownerowi i/lub klientowi (szczególnie wtedy, gdy od pierwszej prezentacji ekranów wprowadzone zostały jakiekolwiek zmiany). Pozwoli to na otrzymanie akceptacji na poszczególnych etapach wdrożenia i unikniemy ewentualnych nieporozumień, które mają tendencję do pojawiania się gdy projekt jest w końcowych fazach produkcji.

Style guide

Omawiając to zagadnienie mamy na myśli dedykowany page lub artboard, gdzie znajdują się podobne do siebie komponenty (np. zestaw kolorów, hierarchia nagłówków, kontrolki formularzy, etc.). Pozwala to na łatwe wychwycenie ich wspólnych właściwości (np. podobieństwo fontów, obramowań i paddingów wewnątrz pól tekstowych, list) i utrzymanie spójności w projekcie. Osoba wdrażająca nasze projekty będzie wiedziała jak ma wyglądać zwykły input w różnych stanach (pusty, z treścią, z placeholder'em, podświetlony kursorem, aktywny, zablokowany) i w różnych wariantach (zwykły input, input do wprowadzania kwoty w danej walucie, dropdown, etc.)

Dodatkowym atutem styleguide'u jest możliwość korzystania z niego jak z pudełka z klockami Lego – nowe ekrany tworzy się szybciej i pewniej (mając świadomość zachowania spójności z resztą projektu). W ekstremalnych przypadkach (np. nasza nieobecność) programista zaopatrzony w dobrze przygotowany style guide będzie sam w stanie zbudować brakujący moduł lub widok z posiadanych komponentów.

Popularne elementy

Nie jesteśmy w stanie przewidzieć i opisać tutaj wszystkich wariantów elementów jakich będziemy potrzebować (przygotowujemy projekty dla różnorodnych rynków i odbiorców), ale z doświadczenia wiemy, że jest kilkanaście typów, które powtarzają się na przestrzeni różnych projektów. Potraktuj tę listę jako inspirację w poszukiwaniu elementów, które mogły Ci umknąć podczas projektowania.

Must have – Jeśli opisywany element da się zastosować w przygotowywanym projekcie, jego przygotowanie jest wymagane.

Nice to have – Jeśli opisywany element da się zastosować w przygotowywanym projekcie, jego przygotowanie jest mile widziane.

KOLORY

Zdefiniuj paletę kolorów. Przygotowując projekt zdefiniuj podstawową paletę kolorów (tło strony, body text, akcenty) i staraj się nimi operować na przestrzeni projektu. Zbyt duża liczba kolorów i ich odcieni może łatwo i szybko wprowadzić wizualny harmider.

Definiuj nowe kolory zamiast zmieniać opacity. Dobrą praktyką jest definiowanie nowych odcieni i unikanie zmian **opacity/alpha** poszczególnych obiektów lub warstw w celu odtworzenia nowych kolorów (chyba, że staraymy się uzyskać efekt przezroczystości danego elementu). Ułatwi to pracę developerom, którzy nie muszą sprawdzać ustawień koloru w conajmniej 3 różnych miejscach (kolor obiektu, przezroczystość obiektu, przezroczystość grupy).

Przygotuj zestaw kolorów dla wykresów. Jeśli tworzony przez Ciebie projekt posiada wykresy, zdefiniuj zestaw kolorów (i kolejność w jakich powinny być używane). Staraj się unikać różnych odcieni tego samego koloru, ponieważ obniża to czytelność wykresu (odbiorca z nawet drobną wadą wzroku może mieć

spore problemy z ich odczytaniem).

NAGŁÓWKI

Zaprojektuj strukturę nagłówków H1-H6. Poza zdefiniowaniem fontu, rozmiaru i koloru pamiętaj również o takim ustawieniu line height / marginesów, aby dobrze komponowały się z pozostałymi nagłówkami i body text.

FORMULARZE

Kontrolki formularzy (form elements):

- Etykieta (Label)
- Pole tekstowe (Input), Pole numeryczne
- Dropdown (Select)
- Textarea
- File input
- Time picker, Date picker, Date time picker
- Suwak / Zakres (Range)
- Button
- Radio button
- Checkbox / Switch
- Opis pola (np. informacja o minimalnej długości hasła)
- Komunikat błędu
- Informację o obowiązku wypełnienia/możliwości pozostawienia pustym (Required/Optional)

Przygotowanie komponentów pól. Poza samym zaprojektowaniem kontrolek formularzy warto zaprojektować też całe pola (np. złożone z etykiety, pola tekstowego i opisu) tak, aby były elastyczne (można było je rozciągać/zwężać) i zawierały zdefiniowane marginesy/odstępy.

Sekcje formularzy. W przypadku długich/skomplikowanych formularzy, podział ich na sekcje (z nagłówkami i opisami) pozytywnie wpływa na szybkość i skuteczność ich wypełniania.

Układ horyzontalny. Niektóre krótkie formularze (takie jak logowanie w nagłówku strony, czy zapisywanie się do newslettera) można ułożyć poziomo (pola są wtedy obok siebie).

SZABLONY E-MAIL

Rozważ wykorzystanie gotowych szablonów email. W przypadku e-maili należy mieć świadomość, że ograniczenia różnych klientów pocztowych wyświetlać je nieco inaczej (co wynika z ich ograniczeń) – część z nich radzi sobie z responsywnymi widokami i GIFami, a inne używają uproszczonej wersji Word’a do renderowania treści (serio!).

Przed przystąpieniem do projektowania szablonów e-mail od zera warto rozważyć tylko drobne zmiany (np. kolorów) w szablonach wbudowanych we framework (np. w przypadku Laravela), skorzystanie z gotowych, darmowych szablonów (np. w [Maizzle](#), który pokochają developerzy uwielbiający Tailwind CSS) lub namówienie klienta na wydanie ok. \$20 na [gotowy, komercyjny szablon](#).

Ogólny szablon dla mailingów transakcyjnych. Warto przygotować ogólny widok, który będzie zawierał:

- Logo
- Kolory tła
- Kilka poziomów nagłówków (zwykle H1-H3 wystarczają)
- Body text
- Hiperłącza
- Sposób ułożenia grafik (rozmiar, marginesy)
- Cytat blokowy

Domyślne fonty. Jeśli nasz projekt korzysta z niestandardowych fontów, dla samych e-maili warto wybrać krój, który jest domyślnie dostępny.

Tytuły maili. Sam projekt graficzny i treść maila to nie wszystko – pamiętaj o przygotowaniu tytułu wiadomości.

STRONY STATYCZNE (BOGATE W TEKST)

Przygotuj jeden gotowy szablon strony statycznej, którą będzie traktowana za uniwersalną (do której będziemy wlewali tekst).

Do tego typu stron można zaliczyć regulamin, politykę prywatności, blog czy strony pomocy. Warto przygotować ogólny widok, który będzie zawierał:

- Nagłówki (H1-H6) ze zdefiniowanymi fontami, kolorami, wysokością wierszy i odstępami przed/po. Warto przygotować też wariant obrazujący wygląd gdy nagłówek jest linkiem.
- Body text
- Hiperłącza
- Sposób ulokowania grafik (rozmiar, marginesy)
- Cytat blokowy
- Listy (uporządkowane i nieuporządkowane)
- Podczas projektowania warto pamiętać o przygotowaniu wariantów responsywnych (w zależności od potrzeb i wytycznych projektu)
- Korzystanie z gotowych stylów dostępnych w bibliotekach (Bootstrap, Tailwind CSS)

STRONY Z UWIERZYTELNIANIEM

Do tego typu stron można zaliczyć:

- Rejestrację nowego konta
- Rejestrację z zaproszenia innego użytkownika (w celu dołączenia do konta firmowego)
- Logowanie + wersja dla włączonego uwierzytelniania dwuskładnikowego (2FA)
- Resetowanie hasła
- Usuwanie konta

ONBOARDING

Omów z klientem/PMem potrzebę przygotowania procesu onboarding.

Istniejący i dobrze przygotowany proces onboardingu to często kluczowy element, który decyduje o tym czy użytkownik będzie korzystał z aplikacji, czy ją porzuci.

Zdefiniuj cele onboardingu. Cele, które będą chcieli osiągnąć stakeholderzy, będą się różniły w różnych aplikacjach. Skomplikowane aplikacje powinny prowadzić użytkownika za rękę pokazując mu kolejne funkcje, wykorzystać kreatory do wypełnienia pustego workspace'a danymi, etc. (Facebook stawia sobie za cel dodanie 4 osób do znajomych w ciągu pierwszych 20 dni od założenia konta; Slack chce, aby wewnątrz firmy wysłano co najmniej 5000 wiadomości).

HAPPY PATH I UNHAPPY PATH

Projektując ścieżki użytkowników warto zastanowić się co powinno się wydarzyć (i jak powinny wyglądać ekrany), gdy dana akcja się nie powiedzie (np. nie udało się zapisać do newslettera ponieważ e-mail jest niepoprawny, lub znajduje się już na naszej liście).

STRONY BŁĘDÓW

Zrozumiałe komunikaty. O ile Błąd 404 przeszedł już do popkultury, tak inne kody błędów (403: Brak uprawnień (Forbidden), lub 500: Błąd serwera (Internal server error) są niezrozumiałe dla większości użytkowników.

Nie pozostawiaj użytkownika samego. Poza opisaniem problemu w sposób zrozumiały dla wszystkich warto jest dać użytkownikom możliwość kontynuowania (np. “Przejdź do strony głównej”, “Napisz do nas na support@companyname.com, żebyśmy mogli Ci pomóc”) zamiast pozostawiania ich w ślepym zaułku.

Dodatkowe materiały

Poza samymi plikami graficznymi, na projekty składają się również:

ZDJĘCIA I ILUSTRACJE

Licencje i prawa do wykorzystania. Upewnijmy się, że posiadamy należyte licencje, które pozwalają nam na korzystanie z nich w danym polu eksploatacji.

Poza osadzeniem takiej grafiki w projekcie przekazujemy je również jako osobne pliki. Jeśli na grafiki w projekcie nakładaliśmy filtry lub efekty, to reprodukujemy je na plikach w wysokiej rozdzielczości i dopiero w takiej formie eksportujemy a następnie przekazujemy.

Zdjęcia i ilustracje przekazujemy nieprzycięte, w maksymalnie wysokiej rozdzielczości. Ułatwi to osobie je wdrażającej ich dopasowanie do różnych formatów, rozdzielczości i gęstości (pikseli) ekranów.

Nie oszczędzamy na wadze plików. Na koniec lepiej kompresować niż denerwować się na niską jakość. Na koniec lepiej kompresować niż denerwować się na niską jakość.

Formaty eksportu grafik.

- Zdjęcia przekazujemy jako JPEG (z najmniejszą możliwą kompresją)
- Ilustracje wektorowe i ikony przekazujemy jako SVG lub ewentualnie EPS,
- Grafiki rastrowe przekazujemy jako PNG 24bit non-interlaced (bez przeplotu) z włączonym transparency (przezroczystość/kanał alpha).
- Eksportując grafiki do docelowych rozmiarów warto zastanowić się nad nowożytnymi formatami kompresji: WEBP, HEIC/HEIF, AVIF, APNG (tam gdzie da się je zastosować)

LOGOTYPY

Przekazujemy wszystkie warianty w popularnych formatach. Jeśli otrzymaliśmy logotypy od klienta, upewnijmy się, że są one w formatach z którymi poradzi sobie frontendowiec (łatwiej poradzić sobie z SVG lub EPS niż z AI).

Przekaż księgę znaku. Jeśli posiadamy księgę znaku lub brandbook (które zwykle zawierają informacje o kolorach, obszarach ochronnych i przykłady

poprawnego/błédnego użycia) – przekażmy je razem z logotypami.

FONTY

- Upewnijmy się, że posiadamy należyte licencje, które pozwalają nam na korzystanie z nich w danym polu eksploatacji.
- Jeśli licencja lub koszty nie pozwalają nam na wykorzystanie danej rodziny czy kroju w projekcie, poszukajmy darmowego lub tańszego zamiennika na Google Fonts, Font Squirrel, czy Adobe Typekit (płatny)
- W przypadku problematycznych/egzotycznych formatów plików możemy spróbować dokonać konwersji (o ile pozwala nam na to licencja)
- W przypadku projektów dla www starajmy się ograniczyć liczbę rodzin i krojów na poszczególnych stronach – poprawi to czytelność i zredukuje rozmiar strony (co przełoży się na szybkość jej ładowania i wyniki w Lighthouse).
- Jeśli to możliwe ograniczajmy charsety (np. jeśli planujemy użyć jakiś specjalny font monospaced do wyświetlania danych numerycznych, nie będziemy potrzebowali glyphów z cyrylicy)

IKONY

Przygotowując projekt www zadbajmy o faviconę (w dobie rosnących rozdzielczości i [szybko rozwijających się przeglądarek](#) warto zastanowić się nad użyciem SVG) i zestaw ikon (w odpowiednich rozmiarach) dla aplikacji PWA.

[Zbiór dokładnych informacji.](#)

Jeśli w projekcie korzystamy z gotowego zestawu ikon, oprócz jego przekazania (np. w formie paczki plików lub linku do biblioteki dostępnej online) warto nazwać komponenty zawierające ikony tak jak nazwy plików (lub dodawać te nazwy w formie komentarzy) – ogromnie ułatwia to znajdowanie i osadzanie poprawnych ikon.

SOCIAL MEDIA

Nawet jeśli klient nie planuje obecności w mediach społecznościowych (w celach stricte

marketingowych), niektóre platformy wymagają założenia kont w celu integracji z ich API.

Open Graph Images. Większość sieci społecznościowych i komunikatorów wyświetla metadane i miniaturę po wklejeniu linku – z tego powodu warto przygotować dedykowane grafiki dla poszczególnych platform (np. Facebook, Twitter). Nie podajemy konkretnych rozdzielczości i formatów, bo te ulegają ciągłym zmianom. Niektóre formaty wymagają odpowiedniego przemyślenia i zaprojektowania, ponieważ ta sama grafika jest przycinana na różne sposoby i wyświetlana na różnych urządzeniach (desktop/mobile).

- <https://www.bannerbear.com/>

Avatary. Część platform społecznościowych operuje awatarami w kształcie kwadratu (zwykle z zaokrąglonymi narożnikami), a część przycina je do kształtu koła. Jeśli niemożliwym jest przygotowanie uniwersalnego obrazka lepiej przygotować dwa osobne. Rozdzielczość **1024×1024** powinna być wystarczająca. Eksportuj je w formie kwadratu, bez przezroczystości (nawet te okrągłe – platformy i aplikacje zawsze same maskują). Dzięki temu nie będzie problemu gdy okrągły avatar (np. z Twittera) zostanie później zamaskowany z użyciem tzw. **Squircle** (jak to miało ostatnio miejsce w Clubhouse, które importowało avatary z Twittera).

POZOSTAŁE

Jeśli wiemy, że przygotowywana strona będzie korzystała z dobrodziejstw PWA, warto przygotować **splash screen** i **ikony aplikacji**.