

## **Przewodnik do CHECKLISTY MAKIETY**

### **1. Dane ogólne (Wykonawca)**

Cel: szybka identyfikacja makiety.

Co wpisać: tytuł, ID scenariusza, obszar tematyczny, rodzaj materiału, data przekazania, zespół (role).

Zalecenie: dopisz platformę docelową i wersję technologii (ułatwia testy).

### **2. Cel makiety (ORE – meryt., techn., WCAG)**

Jak oceniamy: ORE zaznacza trzy checkboxy w pkt 2 dopiero wtedy, gdy spełnione są punkty 3–9. Tu nic nie dopisuje Wykonawca – to zbiorcza akceptacja na podstawie poniższych punktów.

### **3. Zakres funkcjonalny makiety (Wykonawca / ORE – techn., WCAG)**

Minimum, które musi „klikać”:

Nawigacja: Start → Instrukcja → min. 1–2 Ćwiczenia/Zadania → Podsumowanie.

Interakcje: 1–2 działające aktywności z walidacją i feedbackiem (✓/✗/podpowiedź).

UI: finalny szkic layoutu (mogą być assety robocze).

WCAG – podstawy: kontrast, etykiety/alternatywy/ARIA, klawiatura.

Opis funkcjonalności: 1–2 strony „co robią ekrany i przyciski”.

Przykład V.06: Ćw. „buduj z planu”: przenieś 3–5 elementów na planie; zatrask; Podpowiedź 1 = obrys; feedback ✓/✗.

Przykład III.11: Ćw. „siatka anatomiczna”: ułóż 5–7 komórek w siatce organu; walidacja + podpowiedź konturów.

### **4. Warstwa wizualna i UX (Wykonawca / ORE – meryt., WCAG)**

Co pokazać:

Schemat ekranów (mapa nawigacji).

Styl: typografia, kolory, ikonografia, animacje (z grubsza).

3–5 kluczowych ekranów: start, ćwiczenie, feedback, pomoc/wyniki.

WCAG w UI: klik/hitbox  $\geq 44 \times 44$  px, kontrast tekstu  $\geq 4.5:1$ , widoczny focus.

Przykład V.06: trzy widoki obok siebie (plan/przekrój/3D) + wspólne podświetlanie elementów.

Przykład III.11: siatka organu + panel bazy komórek; feedback nad siatką.

## 5. Warstwa techniczna (Wykonawca / ORE – techniczny)

Dostarcz:

- Technologia i silnik (np. React + Canvas / Unity / WebGL / HTML5 / Three.js / Flutter oraz wersje).
- Wersjonowanie: numer wersji również w metadanych komponentu (np. w `engine.json`), zgodny z tagiem/commit w repo.
- Komponent ZPE (repo): URL repo + tag/commit, punkt wejścia silnika (np. `entry.js`), punkt wejścia edytora (jeśli konfiguracja), zasoby stałe, README (jak dodać do materiału i co kliknąć), dane domyślne/demo (opcjonalnie), krótki opis logiki, test kompatybilności z edytorem ZPE (2–3 zdania/screen), obsługiwane urządzenia.
- Smoke test w edytorze – musi przejść: komponent na liście; podgląd działa (interakcje/przejścia); (jeśli jest) panel edytora zapisuje; (jeśli jest) zapis/odtworzenie stanu; (jeśli jest) walidacja; brak zewnętrznych URL-i.

## 6. Zgodność z wymaganiami scenariusza (Wykonawca / ORE – meryt., WCAG, techn.)

Jedna tabela (1 strona):

- Cel ogólny/szczegółowy → ekran/aktywność w makiecie,
- mechanika/interakcja, feedback/podpowiedź,
- WCAG – co wdrożone,
- uwagi/odstępstwa (i kiedy domykamy – np. beta).  
Załącz: mapę przejść (diagram/tekst).

Przykład V.06 (wycinek): „rozpoznaje elementy budowli w planie/przekroju/3D” → ekran „utrwalanie” (3 widoki) → mechanika: wspólne podświetlenie → feedback: nazwa/ikona → WCAG: kontrast, focus.

Przykład III.11 (wycinek): „łączy budowę z funkcją tkanek” → siatka + baza komórek → drag and drop z walidacją → ✓/✗ + zarys → WCAG: etykiety, klawiatura.

## 7. Dokumentacja towarzysząca (Wykonawca)

Dołącz:

- Opis interakcji (1–2 strony),

- User-flow (diagram/lista),
- Lista ekranów + zrzuty (z krótkimi podpisami),
- Lista funkcji do bety (świadome odroczenia - rzeczy, które *na pewno* zostaną pokazane w becie).
- Raport z testów (raport wewnętrzny QA - testy kompatybilności z platformą ZPE + przeglądarki/OS + znane ograniczenia, w przypadku konfliktów należy je opisać oraz propozycje naprawy)

## 8. Raport dla ekspertów ORE (*Wykonawca / ORE*)

Krótko i rzeczowo:

- Opis wykonanych prac i decyzji projektowych,
- Różnice vs scenariusz (jeśli są i uzyskały akceptację ORE) + uzasadnienie dydaktyczne,
- Propozycje do bety (rekomendacje wymagające akceptacji ORE + uzasadnienie dydaktyczne, podane w formie wariantów do decyzji A/B). Przykład:
  - A: dodać 3-stopniowe podpowiedzi (lepszą krzywą nauki, +1 tydz.),
  - B: jedna „super-podpowiedź” (szybciej, ale mniejszy efekt dydaktyczny).  
Rekomendacja wykonawcy: A. Decyzja należy do ORE.
- Pytania do ORE (tematy, które wcześniej nie były poruszane).
- Link do repo + tag/commit oraz 1 screen z edytora (lista komponentów + wstawiony komponent w podglądzie).

## 9. Harmonogram i odpowiedzialność (*Wykonawca*)

Tabelka z datami i rolami: szkice UI → implementacja interakcji → testy WCAG/kompatybilności (QA) → weryfikacja wewn. → przekazanie ORE.

Tip: przy każdym zadaniu wpisz artefakt (np. „UI: 5 ekranów + mapa przejść”).

To jest uszczegółowienie harmonogramów, które zostały dostarczone do 3 miesiąca trwania projektu, spójny z tagowaniem wersji (makieta/beta/final).

## **10. Kryteria odbioru makiety (ORE)**

**Twarde warunki na TAK (wynikają z pkt 3–8):**

- **Finalny szkic UI i kluczowe funkcje działają „na klik”,**
- **Zgodność ze scenariuszem (tabela + mapa przejść) i wymaganiami technicznymi,**
- **Działający prototyp (min. 1–2 aktywności z walidacją i feedbackiem),**
- **Test kompatybilności z ZPE (krótko/screen),**
- **WCAG 2.1 AA – podstawy wykazane,**
- **Komponent uruchamia się w edytorze ZPE i działa podstawowa konfiguracja (lista + „przeciągnij i upuść”),**
- **Podgląd działa, przejście przez 1–2 aktywności możliwe (z komunikatami),**
- **Repo + README dołączone (URL + tag/commit) wersja zgodna z numerem w metadanych.**