

Testovací strategie – Dovolena-za-benefity

Verze dokumentu: 0.0.1

Časový rámec: Strategie stavěná na **max. 4 hodiny práce**, což odpovídá polovině dostupného QA budgetu.

Úvod

Cílem této testovací strategie je pokrýt základní funkční, technické a uživatelské scénáře webu *Dovolena za benefity* na testovací verzi prostředí <https://test-fe-cz.dovolena-za-benefity.cz/> a <https://test-fe-cz.dovolena-za-benefity.cz>

- testování hlavního uživatelského flow,
- stabilitu a funkčnost kritických částí aplikace,
- možnost budoucího rozšíření FE/ E2E automatizace (Playwright),
- jednoduchou opakovatelnost testů pro regresí.

Testování bude probíhat **manuálně i automatizovaně**.

Scope

- Manuální testy
 - Funkční
 - UI/UX
 - Cross-browser
 - Hraniční
 - Exploratory
- Automatizované testy
 - E2E Playwright testy
 - JS/TS
- Dokumentace
 - Regresní testy
 - Automatizované testy
 - Testovací strategie
 - Uživatelské příručky
 - Přístupnost

Out of scope

Lze zavést do budoucna

- Accessibility testy - Lighthouse, WAVE, Axe dle potřeby
- Penetrační testy (ruční XSS, OWASP ZAP, OWASP Top 10, kontrola hlaviček)
- Playwright integrace s API
- Performance testy (např. JMeter) - případné budoucí rozšíření

Manuální testy

Nástroje - Jira/ClickUP = bug tracking, confluence dokumentace či jiný, console, kibana či jiný, browserstack, FakeFiller a další.. (Dle potřeby aplikace)

Desktop prohlížeče

- Chrome (poslední verze)
- Firefox
- Edge
- Safari (macOS)

Testovací scénáře:

- **Funkční:** načtení URL, navigace, validace formulářů
- **Cross-browser:** správné vykreslení prvků napříč prohlížeči
- **UI:** zarovnání, typografie, kontrast
- **Interakce:** scrollování, výběry, kliknutí
- **Hraniční:** dlouhé či nevalidní vstupy

4.2. Mobilní zařízení

- iPhone (Safari)
- Android (Chrome)
- Kontrola responsivního chování: hamburger menu, inputy, scroll, font-size, CTA, modální okna.

4.3. Tablety

- iPad (Safari)
- Android tablet (Chrome)

Testy spíše vizuální + interakce.

Automatizované testy (Playwright – E2E)

3.1. Framework

- **Playwright** (JS/TS)
- Možnost do budoucna propojit s REST API

3.2. Minimální E2E pokrytí

1. **Otevření a ověření webu**
2. **Navigace**
3. **Pozitivní testy např: platný booking.com link**
 - Vložit např. <https://www.booking.com/hotel/...>
 - Odeslat formulář.
 - Ověřit správnou odezvu (např. shrnutí dat / přechod na další krok).
4. **Negativní test – špatná URL**
 - Např: Nesmyslný text.
 - URL z jiného webu (např. Airbnb).
 - Prázdné pole.
5. **Kontrola validací**
 - UI validace.
 - Technická validace (např. skryt/triggery error elementů).
6. **Responsivita (Playwright.devices)**
 - Test minimálně pro iPhone, Pixel a běžný desktop.

3.3. Výstupy

- Automatický HTML report.
- Screenshoty při failu.
- Video z testů (Playwright trace viewer).

Shrnutí

1 měsíc - stabilizace základního testovacího flow:

- Seznámení se s Aplikací
- Dokumentace aplikace - regresní, HDS, AS a další dle potřeby
- Manuální testy kritických funkcí (desktop + mobil).
- Manuální Cross-browser vizuální testy (desktop).
- Manuální testy - negativní scénáře, hraniční hodnoty a další..
- Předávání projektu
- Nastavení Playwright E2E frameworku.
- Konzultace s vývojem

1-3 měsíce - Rozšíření

- Automatizované testy pro hlavní flow.
- Konzultace s vývojem a vedením nad Playwright testy a jejich sepisování
- Konzultace s DEV a nastavení CI/CD např. Jenkins
- Předání prvotních testů vývoji a společné nasazení na CI/CD
- Automatizované spouštění oproti test či prod. Prostředí

3-6 měsíce - Pokračování a vylepšování

- Rozšiřování e2e testů
- Oprava testů které byly rozbity při větších či menších změnách v APP
- Stabilizace a servise
- Rozšíření e2e testů s přidavkem volání API
- Průběžné sledování trendů a nových updatů Playwright či optimalizace

Otázky/odpovědi

- **Rozdělení mezi manuálními a automat. Testy** - viz. Test strategie.
 - Na začátku více manuální testy z důvodu seznámení se s týmem, aplikací a dalšími faktory
 - Do budoucna přechod na automatizace a postupně zvětšovat podíl automatizací (Manuálních testů se nedá obecně všech zbavit)
- **Jak bys zapojil vývojáře do QA procesu?**
 - Kultura je jednoduchá = QA, Vývojář, Produkták = 1 TEAM nic víc nic míň
 - DEV nastavuje a upravuje CI/CD. QA dodává testy a support
 - Sdílení nástrojů, předávání a řešení defektů/bugů
- **AI v QA procesech**
 - Všude kde je to možné a zrychluje process. Není čas ztrácet čas

- Návrhy strategií, dokumentace, UC pro testy atd
 - Pomoc s pathingem, vyhledáváním, nastavováním
- AI je super pomocník ale špatný pán jak se říká. Nutnost tedy vždy kontrolovat. Popřípadně ho pouštět jen tam kam je dovoleno.
- Co bys chtěl mít hotové za 3 a 6 měsíců viz. Shrnutí
- **Whitelabel, multiproduct, jazyky**
 - Používáním XML, CSS či inbuild Playwright. Nebo dle dohody pomocí data-testid. Dále pomocí snapshotů které jsou separovaně odděleny pro jednotlivé testy
 - Co se týče řešení tak jen dvě
 - Každý jazyk má vlastní testovací sadu
 - Modulárně všechny jazyky mají jeden test ale v komandech volám přesný test (Doplním pokud se doberu k zajímavému řešení)