### Strategia Testów w Tribe: Ujednolicenie Wiedzy i Wsparcie Zespołów

Wdrożenie ujednoliconego podejścia do zarządzania wiedzą o testach. Zamiast budowy jednej centralnej platformy, strategia skupia się na **wsparciu zespołów poprzez dostarczenie im gotowych szablonów i potoków (pipeline) CI/CD**, które będą mogli wykorzystać i zaadaptować do własnych potrzeb. Kluczowym elementem jest **ustandaryzowanie wiedzy i stworzenie wspólnej bazy dobrych praktyk**, aby usprawnić współpracę między zespołami i przyspieszyć rozwój kompetencji.

**Kluczowe Korzyści:**

* **Wzrost Produktywności Zespołów:** Dostarczenie gotowych do użycia szablonów i rozwiązań, które eliminują potrzebę "wymyślania koła na nowo".
* **Lepsza Współpraca i Wymiana Wiedzy:** Stworzenie centralnego repozytorium wiedzy i dobrych praktyk ułatwi komunikację i wzajemne wsparcie między zespołami.
* **Podniesienie Jakości i Kompetencji:** Promowanie sprawdzonych wzorców i narzędzi (np. Cypress, Gatling) przyczyni się do podniesienia ogólnej jakości testów i oprogramowania.
* **Większa Autonomia i Elastyczność Zespołów:** Zespoły otrzymują wsparcie i wzorce, ale zachowują elastyczność w doborze narzędzi i procesów najlepiej pasujących do ich specyfiki.

**Inwestycja:** Projekt wymaga powołania grupy roboczej lub Community of Practice, złożonej z ekspertów z różnych zespołów. Ich zadaniem będzie tworzenie i utrzymanie bazy wiedzy, szablonów testowych oraz organizacja wewnętrznych szkoleń i warsztatów. To strategiczna inwestycja w kulturę inżynierską i zdolność organizacji do efektywnego dzielenia się wiedzą.

Spis treści

[Strategia Testów w Tribe: Ujednolicenie Wiedzy i Wsparcie Zespołów 1](#_Toc202954466)

[1. Wprowadzenie i Cele Strategiczne 2](#_Toc202954467)

[2. Stan Obecny: Analiza Procesów Testowych 2](#_Toc202954468)

[Mapa Dojrzałości i Zdolności Testowych 4](#_Toc202954469)

[3. Uzasadnienie Strategii 5](#_Toc202954470)

[4. Wizja Docelowa: Ekosystem Wsparcia dla Testów 6](#_Toc202954471)

[5. Implementacja: Wymagania i Wyzwania 6](#_Toc202954472)

### 1. Wprowadzenie i Cele Strategiczne

#### 1.1. Kontekst Biznesowy

W dynamicznie rozwijającym się środowisku technologicznym szybkość i jakość dostarczanego oprogramowania są kluczowymi czynnikami sukcesu. Obecne zróżnicowanie procesów i narzędzi testowych w poszczególnych zespołach Tribe, mimo że pozwala na realizację bieżących zadań, prowadzi do powstawania silosów wiedzy, problemów ze spójnością środowisk oraz utrudnia efektywne zarządzanie jakością na poziomie całej organizacji.

#### 1.2. Myśl Przewodnia: Wspólna Baza Wiedzy, Lepsze Zespoły

Ten dokument przedstawia strategię transformacji podejścia do testowania. Jej myślą przewodnią jest ujednolicenie i usystematyzowanie wiedzy o testach oraz stworzenie mechanizmów wspierających zespoły. Celem jest przejście od rozproszonych, często nieudokumentowanych praktyk, do modelu, w którym każdy zespół ma dostęp do wspólnej bazy dobrych praktyk, gotowych szablonów i wsparcia ekspertów, co przekłada się na stabilność i jakość ich pracy.

#### 1.3. Główne Cele Inicjatywy

* **Usystematyzowanie i centralizacja wiedzy** o testach, narzędziach i procesach w całym Tribe.
* **Stworzenie bazy dobrych praktyk i szablonów testowych** w celu zwiększenia efektywności i redukcji powielanej pracy.
* **Wsparcie zespołów w rozwiązywaniu problemów** związanych z automatyzacją i środowiskami testowymi.
* **Skrócenie pętli feedbacku** (czasu od zmiany w kodzie do uzyskania wyniku testu) poprzez promowanie efektywnych strategii automatyzacji.
* **Stworzenie jednego, centralnego miejsca** do wymiany wiedzy, ułatwiającego onboarding i rozwój kompetencji testerskich.

### 2. Stan Obecny: Analiza Procesów Testowych

#### 2.1. Ogólne Wyzwania i Obserwacje

Na podstawie wstępnego zebrania informacji zidentyfikowano powtarzające się wyzwania:

* Dokumentacja
* Środowiska
* Automatyzacja

#### 2.2. Szczegółowa Analiza Zespołów

Poniższa tabela przedstawia zagregowaną analizę strategii testowych w zdiagnozowanych zespołach.

|  |  |  |  |
| --- | --- | --- | --- |
| **Zespół** | **Rodzaj Testów** | **Narzędzia i Proces** | **Wnioski** |
| **Płatności** | UI, API (manualne), Wydajnościowe | Nightwatch (przez pipeline), Postman(manualnie), Gatling(lokalnie) | Strategia silnie oparta na testach frontowych. Duże obciążenie pracą manualną hamuje rozwój i pozyskiwanie nowej wiedzy. |
| **MIG HCE Visa Mobile** | Funkcjonalne, Wydajnościowe, Manualne E2E | Cypress (na VDI), Gatling | Problemy ze stabilnością dedykowanego środowiska VDI. |
| **MIG Blik** | Frontowe, Backendowe, Manualne | Nightwatch.js, Cypress. Pełne pokrycie manualne. | Brak testów wydajnościowych. Dobra automatyzacja front/back, ale dokumentacja jest rozproszona. Główny problem: brak czasu. |
| **Ferryt** | E2E (Front), Manualne | Playwright. Testy manualne na wielu platformach. | Tylko 1 z 5 testerów automatyzuje. Brak dedykowanych środowisk, czasu na automatyzację i dokumentację. |
| **SPE1 SPE2** | UI, Wydajnościowe, Mobile | Nightwatch, Gatling, Saucelabs. Testy głównie przez VDI. | Niewielka ilość automatyzacji, potrzeba rozwoju testów backendowych. Dokumentacja w SNOW/TFS. |
| **Konta** | Manualne E2E, Automatyczne (Front), WCAG | Proces wciąż mocno oparty na testach manualnych na UAT. | Brak formalnej dokumentacji (tylko lista przypadków). Trudności z testami WCAG i brak czasu na rozwój automatyzacji. |
| **Planner** | Backend, Frontend, Wydajnościowe | Thunder Client(manualnie), automatyzacja w toku. Gatling. | Duża liczba aplikacji w utrzymaniu. Brak czasu jest główną barierą dla rozwoju automatyzacji. |
| **PFM Keraaja Historia** | Backend, Wydajnościowe, Manualne | Cucumber (dla testów funkcjonalnych), Gatling. | Dobry balans testów w procesie, opartym o środowisko UAT i mocki. |
| **Karty** | Manualne (wszystko), E2E (Front), Regresyjne | Nightwatch, Jmeter, Postman, SoapUI i wiele innych. | Ogromna zależność od testów manualnych. Brak standaryzacji narzędzi. Problemy ze środowiskami i zarządzaniem danymi testowymi. |

### 

### Mapa Dojrzałości i Zdolności Testowych

**Legenda:**

* ● - Tak / Zdolność istnieje
* ◐ - Częściowo / W toku / Proces manualny mimo automatyzacji
* ○ - Nie / Brak zdolności

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Zespół (Squad) | Automatyzacja UI / Front-end | Automatyzacja API / Back-end | Testy Wydajnościowe | Duży udział Procesu Manualnego | Uporządkowana Dokumentacja |
| **Płatności** | ● | ○ | ● | ● | ○ |
| **MIG HCE  Visa Mobile** | ○ | ● | ● | ◐ | ○ |
| **BLIK** | ● | ● | ○ | ● | ○ |
| **Terryt** | ● | ○ | ○ | ● | ○ |
| **SPE1 SPE2** | ● | ○ | ● | ● | ◐ |
| **Konta** | ◐ | ○ | ○ | ● | ○ |
| **Planner** | ◐ | ◐ | ● | ● | ○ |
| **PFM** | ○ | ● | ● | ◐ | ● |
| **Karty** | ● | ○ | ● | ● | ○ |

### 3. Uzasadnienie Strategii

#### 3.1. Dlaczego Wspólna Baza Wiedzy?

Proponowane podejście nie narzuca jednej, sztywnej standaryzacji, lecz skupia się na **stabilności i rozwoju wiedzy wewnątrz zespołów oraz pomiędzy nimi**. Zamiast budować kosztowną platformę centralną, inwestujemy w ludzi i procesy. Ujednolicenie ma dotyczyć dostępu do wiedzy, dobrych praktyk oraz gotowych do użycia narzędzi i szablonów. Takie podejście promuje kulturę **wewnętrznego open source (InnerSource)**, gdzie zespoły mogą czerpać ze wspólnych zasobów i jednocześnie wnosić do nich swój wkład, co naturalnie podnosi jakość i efektywność w całej organizacji.

#### 3.2. Porównanie: Przed i Po Wdrożeniu Inicjatywy

Wprowadzenie ekosystemu wsparcia dla testów przyniesie fundamentalną zmianę w codziennej pracy zespołów.

|  |  |  |
| --- | --- | --- |
| Aspekt | Stan Obecny | Stan Docelowy |
| Wiedza o Testach | Rozproszona, nieudokumentowana, w "głowach" pojedynczych osób. | Scentralizowana, dostępna dla każdego w wspólnej bazie wiedzy (np. Confluence). |
| Start Nowego Projektu/Testu | Każdy zespół tworzy rozwiązania od zera, powielając pracę. | Zespoły korzystają z gotowych szablonów i potoków (pipeline), co przyspiesza start. |
| Współpraca Międzyzespołowa | Utrudniona z powodu braku wspólnego języka i standardów. | Ułatwiona dzięki wspólnym praktykom i regularnym spotkaniom (Community of Practice). |
| Wdrożenie Nowego Testera | Długotrwałe, oparte na nieformalnym przekazywaniu wiedzy. | Szybkie, oparte na jasnej dokumentacji, przykładach i centralnej bazie wiedzy. |

Eksportuj do Arkuszy

### 4. Wizja Docelowa: Ekosystem Wsparcia dla Testów

Celem jest stworzenie wewnętrznego ekosystemu, który wspiera testerów i deweloperów w ich codziennej pracy. Zamiast jednej platformy, wizja opiera się na trzech filarach:

1. **Centralna Baza Wiedzy:** Jedno miejsce (np. strona w Confluence) zawierające:
   * Opisy dobrych praktyk (np. jak pisać stabilne testy E2E).
   * Dokumentację używanych narzędzi i frameworków.
   * Przykłady i tutoriale.
2. **Repozytorium Szablonów i Zasobów:** Centralne repozytorium Git, w którym znajdować się będą:
   * Szablony projektów testowych (np. dla Cypress, Gatling).
   * Gotowe potoki CI/CD (Azure Pipelines) do uruchamiania testów.
   * Współdzielone biblioteki i narzędzia pomocnicze.
3. **Community of Practice (Społeczność Praktyków):** Regularne spotkania (formalne i nieformalne) dla wszystkich osób zaangażowanych w jakość, służące do:
   * Wymiany doświadczeń i rozwiązywania problemów.
   * Prezentacji nowych narzędzi i technik.
   * Wspólnego rozwoju bazy wiedzy i szablonów.

### 5. Implementacja: Wymagania i Wyzwania

#### 5.1. Wymagania

* **Platforma do Dokumentacji:** Dostęp i konfiguracja przestrzeni na wspólną bazę wiedzy (np. Confluence).
* **Repozytorium Kodu:** Skonfigurowane repozytorium Git (np. w Azure DevOps) do przechowywania szablonów i zasobów.

#### 5.2. Wymagane Kompetencje i Rozwój Zespołów

* **Kultura Dzielenia się Wiedzą:** Promowanie otwartości i chęci do dokumentowania swojej pracy oraz pomagania innym.
* **Umiejętności Techniczne:** Rozwój kompetencji w kluczowych narzędziach (np. Cypress, Gatling, Docker) poprzez warsztaty i pracę z szablonami.