**Raport o projekcie metodą Scrum**



**1. Wstęp**



Niniejszy raport przedstawia szczegółowy opis procesu rozwóju projektu „Project Manager” z wykorzystaniem metodologii Scrum. Projekt został pomyślnie ukończony, a zespół osiągnął wszystkie założone cele. Raport zawiera informacje o rolach uczestników, etapach pracy, zrealizowanych user stories, wynikach, refleksji Scrum Mastera, a także wnioskach i rekomendacjach.



**2. Opis projektu**



Projekt „Project Manager” został opracowany w celu stworzenia aplikacji ułatwiającej zarządzanie projektami przez zespoły. Głównym celem było dostarczenie narzędzia do przydzielania ról, wyznaczania zadań, śledzenia postępów oraz wizualizacji stopnia realizacji projektu poprzez procentowy wskaźnik wykonanych zadań. Aplikacja obejmuje wersję mobilną (opartą na .NET MAUI) oraz wersję webową (opartą na Vue.js), zintegrowane z backendem opartym na Java Spring Boot, wdrożonym na serwerze Ubuntu.

Projekt miał na celu automatyzację procesów zarządzania zadaniami, umożliwiając zespołom skupienie się na ich realizacji, a nie na ręcznym monitorowaniu postępów. Kluczową cechą była możliwość monitorowania postępu w procentach, co pozwalało na wizualną ocenę zbliżania się projektu do zakończenia – szczególnie przydatne dla zespołów pracujących w ograniczonym czasie, takich jak projekty uniwersyteckie.



**3. Role w zespole**



Każdy członek zespołu pełnił określone funkcje, co umożliwiło efektywne rozdzielenie zadań i osiągnięcie celów. Poniżej przedstawiono role uczestników:

**Scrum Master**: Anton Katolichenko – koordynował pracę zespołu, prowadził planowanie sprintów, codzienne stand-upy i retrospektywy, rozwiązywał konflikty i przeszkody. Zapewniał przestrzeganie zasad Scrum i wspierał motywację zespołu.



**Zespół deweloperski**:



**Anton Katolichenko** – rozwój aplikacji mobilnej, integracja z API. Skupił się na tworzeniu interfejsów użytkownika i logiki dla wersji mobilnej.



**Hleb Matsehora** – rozwój i konfiguracja backendu, w tym serwera i CI/CD. Odpowiedzialny za infrastrukturę serwerową i bezpieczeństwo API.



**Mark Kandratsiuk** – rozwój frontendu wersji webowej, komponenty Vue.js. Realizował interfejsy i integrację z backendem.



**Ilia Kaliada** – projektowanie UX/UI, wsparcie dla frontendu. Tworzył design aplikacji i zapewniał jego spójność między platformami.



**Ilya Hruzinski** – projektowanie UX/UI, testowanie. Przeprowadzał testy manualne i automatyczne, identyfikował błędy.



Każdy uczestnik wnosił wkład zgodnie ze swoimi umiejętnościami, co zapewniło zrównoważone rozdzielenie obowiązków. Zespół liczył pięć osób, co pozwoliło na elastyczne dostosowanie ról w zależności od potrzeb sprintu.



**4. Proces pracy**



Projekt został zrealizowany w trzech sprintach, każdy trwający około dwóch tygodni. Proces obejmował planowanie sprintu, codzienne stand-upy, rozwój, testowanie i retrospektywy, co pozwoliło zespołowi pozostać zorganizowanym i reagować na zmiany.

**Sprint #1 (16.03–01.04)**

**Cel**: Realizacja podstawowych funkcji, takich jak rejestracja, logowanie, zarządzanie profilem i ustawianie awatara.



**Zadania**:



Konfiguracja infrastruktury serwerowej (Hleb Matsehora): instalacja Jenkins, Docker, PostgresSQL, PgAdmin; konfiguracja projektu w Java Spring Boot z obsługą Swagger.



Tworzenie projektu mobilnego (Anton Katolichenko): inicjalizacja projektu w .NET MAUI, architektura MVVM, integracja bibliotek (CommunityToolkit.Maui, Refit).



Analiza technologii i projektowanie UI (Mark Kandratsiuk, Ilya Kaliada): badanie Vue.js, tworzenie wstępnych makiet w Figma.



Przygotowanie do testowania (Ilya Hruzinski): wybór narzędzi testowych, analiza wymagań.



**Wynik**: Zrealizowano user stories #1–#4. Zespół zyskał podstawowy funkcjonalność, umożliwiającą rozpoczęcie pracy z aplikacją.



**Sprint #2 (01.04–17.04)**

**Cel**: Rozwój funkcji zarządzania projektami i sprintami, w tym tworzenie projektów, dołączanie do nich i zarządzanie rolami.

**Zadania**:



Tworzenie interfejsów dla projektów i sprintów (Anton Katolichenko, Mark Kandratsiuk, Ilia Kaliada): rozwój ekranów dla wersji mobilnej i webowej.



Integracja nowych endpointów w backendzie (Hleb Matsehora): dodanie API do zarządzania projektami.



Testowanie wersji webowej (Ilya Hruzinski): weryfikacja rejestracji i logowania, identyfikacja początkowych błędów.



**Wynik**: Zrealizowano user stories #5–#12. Zespół wdrożył kluczowe funkcje zarządzania projektami, umożliwiając dalszą pracę nad zadaniami.



**Sprint #3 (17.04–23.04)**

**Cel**: Udoskonalenie funkcji zarządzania zadaniami, w tym przydzielanie, statusy i priorytetyzacja.



**Zadania**:



Realizacja zarządzania zadaniami (Anton Katolichenko, Mark Kandratsiuk): tworzenie interfejsów do przydzielania zadań i zmiany ich statusów.



Testowanie i poprawianie błędów (Ilya Hruzinski): pisanie testów dla komponentów, weryfikacja integracji.



Optymalizacja backendu (Hleb Matsehora): dopracowanie endpointów, poprawa wydajności.



**Wynik**: Zrealizowano user stories #13–#18. Projekt zyskał pełną funkcjonalność zarządzania zadaniami, gotową do prezentacji.



Proces pracy był organizowany za pomocą Microsoft Teams, co umożliwiło śledzenie postępów i rozdzielanie zadań. Codzienne stand-upy odbywały się online, co ułatwiło komunikację.



**5. User Stories**



Poniżej przedstawiono wszystkie user stories projektu z krótkim opisem ich realizacji. Wszystkie zostały pomyślnie ukończone.

**User Story #1**: Jako użytkownik chcę stworzyć konto. Zrealizowano poprzez ekran rejestracji w .NET MAUI i Vue.js z walidacją danych i integracją z API.



**User Story #2**: Jako użytkownik chcę się zalogować. Dodano ekran logowania z generowaniem tokena JWT.



**User Story #3**: Jako użytkownik chcę zarządzać profilem. Zaimplementowano ekran profilu z możliwością edycji danych.



**User Story #4**: Jako użytkownik chcę ustawić awatar. Dodano funkcję przesyłania zdjęć z walidacją rozmiaru i formatu.



**User Story #5**: Jako użytkownik chcę tworzyć projekty. Zrealizowano poprzez interfejs z formularzem tworzenia projektu.



**User Story #6**: Jako użytkownik chcę dołączać do projektów. Dodano generowanie kodu dołączenia.



**User Story #7**: Jako użytkownik chcę widzieć listę projektów. Zaimplementowano listę projektów z filtrowaniem.



**User Story #8**: Jako menedżer chcę przydzielać role. Dodano interfejs zarządzania rolami.



**User Story #9**: Jako twórca chcę mianować administratorów. Zaimplementowano funkcję mianowania roli administratora.



**User Story #10**: Jako administrator chcę usuwać członków zespołu. Dodano przycisk usuwania uczestnika.



**User Story #11**: Jako użytkownik chcę tworzyć sprinty. Zrealizowano interfejs tworzenia sprintu.



**User Story #12**: Jako użytkownik chcę tworzyć zadania. Dodano formularz tworzenia zadania.



**User Story #13**: Jako użytkownik chcę przydzielać zadania. Zaimplementowano wybór uczestników dla zadania.



**User Story #14**: Jako użytkownik chcę grupować zadania w sprinty. Zadania powiązano ze sprintami.



**User Story #15**: Jako użytkownik chcę widzieć swoje zadania. Zaimplementowano osobistą listę zadań.



**User Story #16**: Jako użytkownik chcę zmieniać status zadania. Dodano statusy „W trakcie”, „Zakończone”.



**User Story #17**: Jako użytkownik chcę oznaczac zadania jako ważne. Zrealizowano oznaczanie ważności.



**User Story #18**: Jako użytkownik chcę widzieć ważność zadań. Dodano wyświetlanie ważności na liście.



Każda user story została przetestowana zgodnie z kryteriami akceptacji, co zapewniło jakość produktu.



**6. Wyniki**



Projekt został ukończony z pełnym zrealizowaniem wszystkich 18 user stories. Stworzono wersje mobilną i webową aplikacji, zintegrowane z backendem. Część serwerowa została skonfigurowana z wsparciem CI/CD za pomocą Jenkins, co ułatwiło wdrożenie. Prototyp w pełni spełnia wymagania i otrzymał pozytywną ocenę od prowadzącego. Aplikacja umożliwia tworzenie projektów, zarządzanie zadaniami i śledzenie postępów, co czyni ją użytecznym narzędziem do pracy zespołowej.



**7. Testowanie**



**Testowanie przeprowadzono na wszystkich etapach projektu, obejmując zarówno metody manualne, jak i automatyczne: - Testowanie manualne: Ilya Hruzinski zweryfikował funkcjonalność wersji webowej i mobilnej, w tym rejestrację, logowanie i zarządzanie zadaniami. - Testowanie automatyczne: Stworzono testy dla komponentów aplikacji webowej (ProjectDialog.vue, Login.vue, Register.vue) oraz backendu (Hleb Matsehora). Testy obejmowały kluczowe funkcje, takie jak walidacja danych i bezpieczeństwo. - Zidentyfikowane błędy:**

**- B001: Brak nagłówka przy pustym project.name w ProjectDialog.**

**- B003: Brak emisji update:show przy zamykaniu poza oknem dialogowym.**

**- B005: Niedostateczna walidacja pustych pól w formularzu logowania. To część błędów, które zostały znalezione.**



**8. Refleksja Scrum Mastera**



Jako Scrum Master, Anton Katolichenko, regularnie prowadziłem (dokumentowałem) refleksję dla każdego członka zespołu. Codzienne stand-upy pomagały monitorować postępy i szybko rozwiązywać problemy, takie jak opóźnienia w integracji API czy niezgodności w projektowaniu UI. Po każdym sprincie odbywały się retrospektywy, podczas których omawiano sukcesy i obszary do poprawy. Na przykład w sprincie #1 napotkano problem niewystarczającej komunikacji między projektantami a programistami, co opóźniło realizację UI. Zostało to rozwiązane poprzez wprowadzenie dodatkowych spotkań koordynacyjnych.

Kluczowym zadaniem jako Scrum Master było zarządzanie czasem. W sprincie #3 pojawił się problem braku czasu na testowanie, co wymagało redistribucji zadań – część programistów wspierała pisanie testów, co pozwoliło dotrzymać terminów. Wspierałem także motywację zespołu, organizując nieformalne dyskusje i pomagając rozwiązywać konflikty. To doświadczenie nauczyło mnie znaczenia wczesnego planowania testów i precyzyjnego rozdzielania obowiązków.



**9. Aspekty techniczne**



Projekt obejmował kilka komponentów technicznych, które zapewniły jego funkcjonalność:

**Wersja mobilna**: Opracowana w .NET MAUI z wykorzystaniem architektury MVVM. Użyto bibliotek CommunityToolkit.Maui, CommunityToolkit.Mvvm, DevExpress.Maui dla ułatwienia rozwoju.



**Wersja webowa**: Stworzona w Vue.js z użyciem Vuex do zarządzania stanem. Integracja z backendem poprzez REST API.



**Backend**: Zrealizowany w Java Spring Boot. Wykorzystano Spring Security do autoryzacji, Swagger do dokumentacji API, PostgresSQL do przechowywania danych.



**Serwer**: Wdrożony na Ubuntu (Hetzner) z użyciem Docker i Jenkins dla CI/CD. Umożliwiło to automatyzację wdrożeń i testów.



Architektura techniczna została wybrana z myślą o skalowalności i łatwości integracji. Na przykład użycie Swagger pozwoliło na szybkie testowanie API, a Docker uprościł wdrożenie na serwerze.



**10. Rekomendacje**



Na podstawie doświadczeń z projektem można zaproponować następujące rekomendacje dla przyszłych projektów:

Rozpocząć testowanie już w pierwszym sprincie, aby wychwytywać błędy na wczesnym etapie.



Zwiększyć udział testów automatycznych, np. dodać testy z użyciem Selenium dla weryfikacji UI.



Ulepszyć dokumentację API, aby ułatwić integrację między frontem a backendem.



Wykorzystać narzędzia do zarządzania projektami (np. Jira) z bardziej szczegółową konfiguracją, aby poprawić śledzenie zadań.



Prowadzić regularne przeglądy designu pod kątem zgodności z wymaganiami, aby uniknąć niezgodności na etapie rozwoju.



Te kroki pomogą ulepszyć proces rozwoju i podnieść jakość końcowego produktu.



**11. Wnioski i lekcje**



**Sukcesy**: Zespół efektywnie zastosował metodologię Scrum, co pozwoliło ukończyć projekt w terminie. Stworzony prototyp jest w pełni funkcjonalny i spełnia wymagania.



**Trudności**: Główne wyzwania związane były z brakiem czasu na poprawienie wszystkich błędów oraz koordynacją między platformami (wersja mobilna i webowa). Pojawiały się także problemy z integracją API z powodu niewystarczającej dokumentacji.



**Lekcje**: Retrospektywy pomogły poprawić komunikację w zespole. Ważne jest rozpoczynanie testów od pierwszych etapów projektu i dokładne planowanie zadań. Zespół nauczył się także efektywniej rozdzielać role i obowiązki.



**12. Zakończenie**



Projekt „Project Manager” stanowi udany przykład zastosowania metodologii Scrum w projekcie uniwersyteckim. Zespół stworzył funkcjonalny prototyp, który umożliwia efektywne zarządzanie projektami i zadaniami. Zdobyty опыт, w tym umiejętności pracy zespołowej, zarządzania czasem i rozwoju technicznego, będzie cenny w przyszłych projektach. Rekomendacje przedstawione w raporcie pomogą ulepszyć proces rozwoju i uniknąć błędów w przyszłości.



*Data sporządzenia raportu: 22 maja 2025 r., godz. 12:11 CEST.*