

Przygotowanie środowiska pracy

Przygotowanie środowiska pracy to nie tylko instalacja i konfiguracja niezbędnych aplikacji i narzędzi, ale również nauka optymalnego ich wykorzystania. Obecnie wszelkiego rodzaju tutoriale i szkolenia e-learningowe niemal „wylewają się” z internetu, dlatego coraz częściej wymagana jest znajomość wielu narzędzi już na początku pracy — po to by proces tworzenia oprogramowania był jak najszybszy i jak najbardziej wydajny.

1.1. Narzędzia wykorzystywane w prowadzeniu projektu

W dzisiejszych czasach realizacja projektów jest tak różnorodna, że rynek nieustannie dostarcza nowych rozwiązań technologicznych, które mają za zadanie wspomagać procesy i zarządzanie projektami. W niniejszym rozdziale zostaną omówione narzędzia (zarówno płatne, jak i darmowe), które wspierają zarządzanie projektem, zarządzanie testami oraz wytwarzanie oprogramowania.

UWAGA

Rozdział ma charakter wprowadzający — warto mieć na uwadze, że niżej wymienione narzędzia cały czas są rozwijane i wzbogacane o kolejne funkcje.

1.1.1. HP ALM

Jako pierwsze zostanie omówione narzędzie HP ALM. Jest to aplikacja do wspierania procesu testowego i zarządzania testami. Dostęp do narzędzia jest przyznawany na podstawie licencji (1 licencja = 1 użytkownik). Umożliwia ono planowanie oraz wykonanie testów manualnych i automatycznych, zarządzanie defektami (zgłaszanie, monitorowanie). HP ALM dzieli się na dwie części:

- administracyjną (ang. *Site Administration*);
- użytkownika (ang. *ALM Platform*) — cały moduł przeznaczony dla użytkowników.

W dalszej części rozdziału skupimy się na części użytkownika, gdyż panel administracyjny jest dostępny — jak sama nazwa wskazuje — dla osób pełniących funkcję administratora. W panelu administracyjnym wykonuje się wszystkie czynności administracyjne, takie jak tworzenie domen, projektów, użytkowników oraz nadawanie uprawnień.

Po wybraniu adresu docelowego narzędzia w danej organizacji pojawia się ekran logowania (rysunek 1.1):

Rysunek 1.1. Ekran logowania w HP ALM

Jak można zauważyć, poza wymaganymi danymi do logowania aplikacja wymusza na nas wybór z list rozwijanych domeny (ang. *Domain*) i projektu. Użytkownik może się zalogować wyłącznie do projektów, do których nadano mu dostęp.

UWAGA

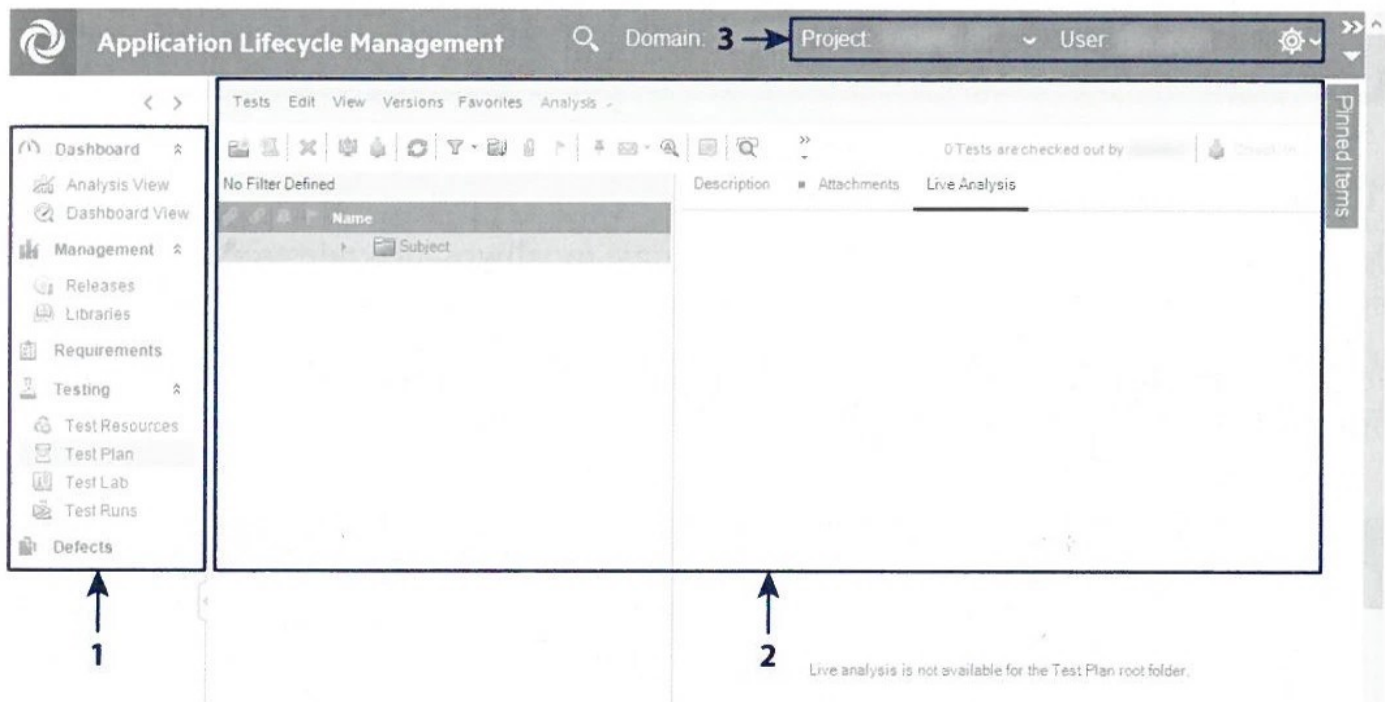
Po zalogowaniu widoczne są tylko te funkcje, do których mamy dostęp.

Panel użytkownika może się składać z następujących modułów (widoczność zależna od uprawnień):

- **Dashboard** — statystyki, raporty;
- **Management** — zarządzanie planem wersji;
- **Requirements** — definiowanie wymagań systemowych i zarządzanie nimi;
- **Testing** — tworzenie zestawów testów w projektach, uruchamianie zaplanowanych testów;
- **Defects** — tworzenie i śledzenie statusów defektów.

Poniżej przedstawiono podgląd widoku ekranu po zalogowaniu do aplikacji (rysunek 1.2). Widać na nim trzy grupy menu:

1. Menu wyboru modułów.
2. Menu operacji dostępnych w wybranym wcześniej module.
3. Menu użytkownika (wybór projektu, ustawienia konta).



Rysunek 1.2. Widok na menu i moduły w HP ALM

W dalszej części podrozdziału przedstawione zostaną moduły *Testing* i *Defects*.

1.1.1.1. Moduł Testing

Jest to moduł służący do organizowania procesu testowego. Składa się z następujących elementów (użytkownik widzi tylko te, do których ma nadany dostęp):

- *Test Resources* — zawiera wszystkie dodatkowe zasoby używane do testów manualnych oraz automatycznych, zarówno funkcjonalnych, jak i wydajnościowych.
- *Business Components* — moduł, który pozwala zarządzać komponentami biznesowymi.
- *Test Lab* — tu przenosimy testy, które są skonfigurowane i gotowe do uruchomienia.
- *Test Runs* — w module widoczne są wszystkie uruchomienia testów wraz z ich wynikami.
- *Test Plan* — moduł pozwalający na zorganizowanie całego procesu testowania poprzez określenie, jakie testy zostaną wykonane w celu zweryfikowania poprawności działania systemu.

DEFINICJA

Weryfikacja — egzaminowanie poprawności i dostarczenie obiektywnego dowodu na to, że produkt procesu wytwarzania oprogramowania spełnia zdefiniowane wymagania.

Walidacja — sprawdzenie poprawności i dostarczenie obiektywnego dowodu na to, że produkt procesu wytwarzania oprogramowania zaspokaja potrzeby użytkownika i spełnia jego wymagania.

1.1.1.2. Moduł Defects

Jest to moduł do zarządzania błędami, który daje możliwość ich rejestrowania, przypisywania do konkretnych osób i zmiany statusów zarejestrowanych błędów.

Możemy zgłosić tutaj wszystkie uwagi do zachowań (zarówno funkcjonalnych, jak i niefunkcjonalnych) systemu odbiegających od przyjętych wymagań.

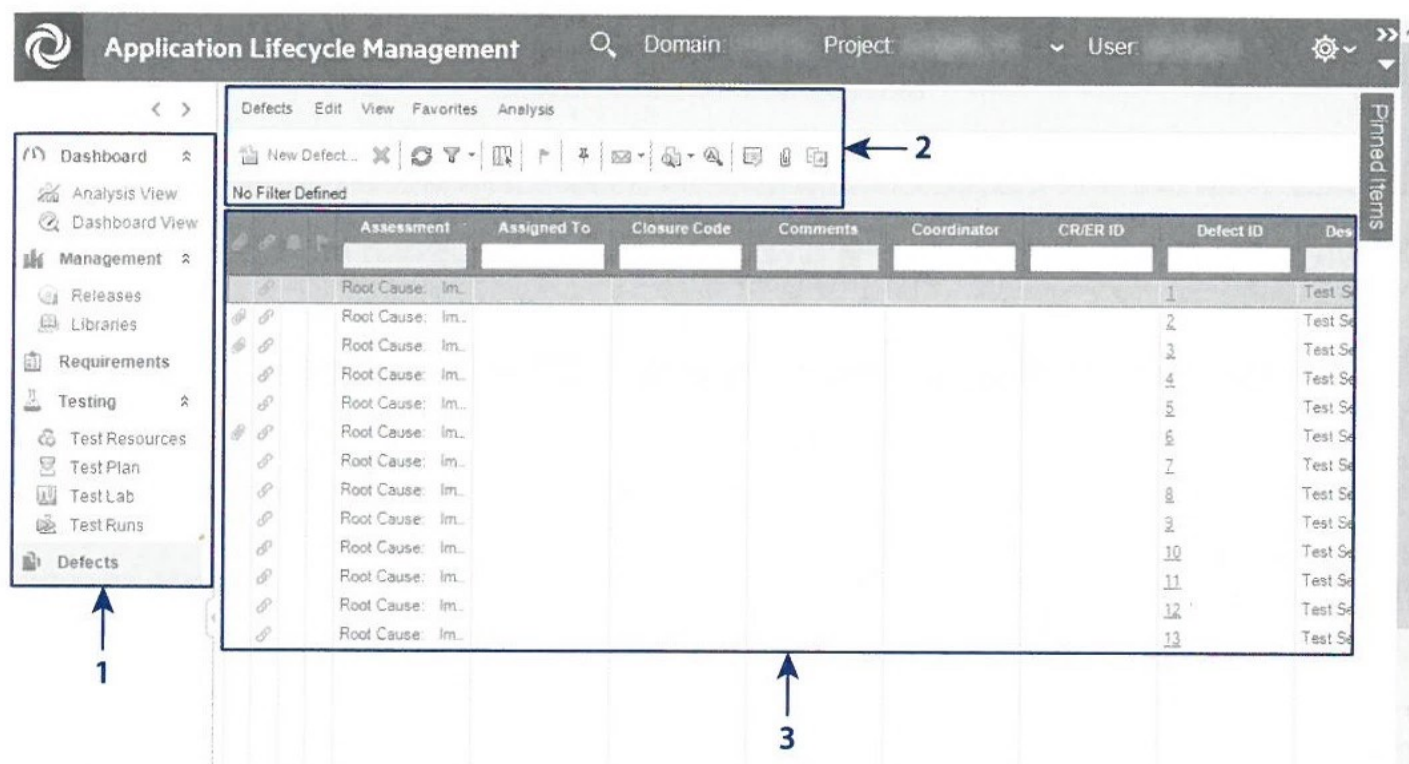
DEFINICJA

Wymaganie funkcjonalne dotyczy wyniku zachowania systemu, który powinien zostać dostarczony przez funkcję systemu.

Wymaganie niefunkcjonalne dotyczy jakości tworzonego oprogramowania, ale nie zostało określone przez wymagania funkcjonalne. Odnosi się m.in. do wydajności, dostępności, niezawodności, skalowalności i przenośności.

Poniżej przedstawiono podgląd widoku ekranu modułu *Defects* (rysunek 1.3). Można na nim zauważyć trzy grupy menu:

1. Menu wyboru modułów.
2. Menu górne.
3. Widok na wszystkie zgłoszenia w ramach projektu, do którego jesteśmy zalogowani.



Rysunek 1.3. Widok narzędzia w module Defects

Defekty możemy zgłosić zarówno z poziomu modułu *Test Lab*, jak i z poziomu *Defects*. W menu górnym (2) należy najpierw wybrać *New Defect*, a następnie wypełnić wszystkie wymagane pola w formularzu. Po dodaniu defektu jest on widoczny na liście (3).

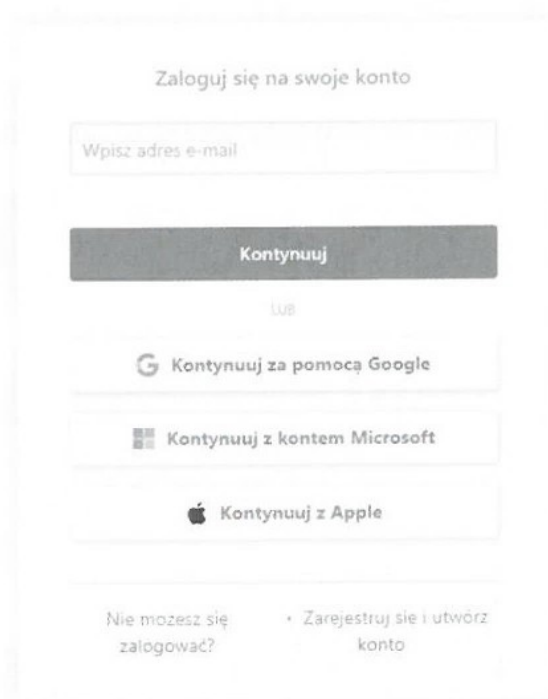
HP ALM jest bardzo elastycznym narzędziem. Możemy je spersonalizować do każdego projektu. Co ważne — możemy sami definiować raporty czy widoki i udostępniać je innym osobom.

Jest to jedno z najdroższych narzędzi na rynku, jednak jest warte swojej ceny.

1.1.2. Jira

Jira jest komercyjnym narzędziem nie tylko wykorzystywanym w zarządzaniu projektami, ale również wspomagającym elektroniczny obieg dokumentacji w firmie.

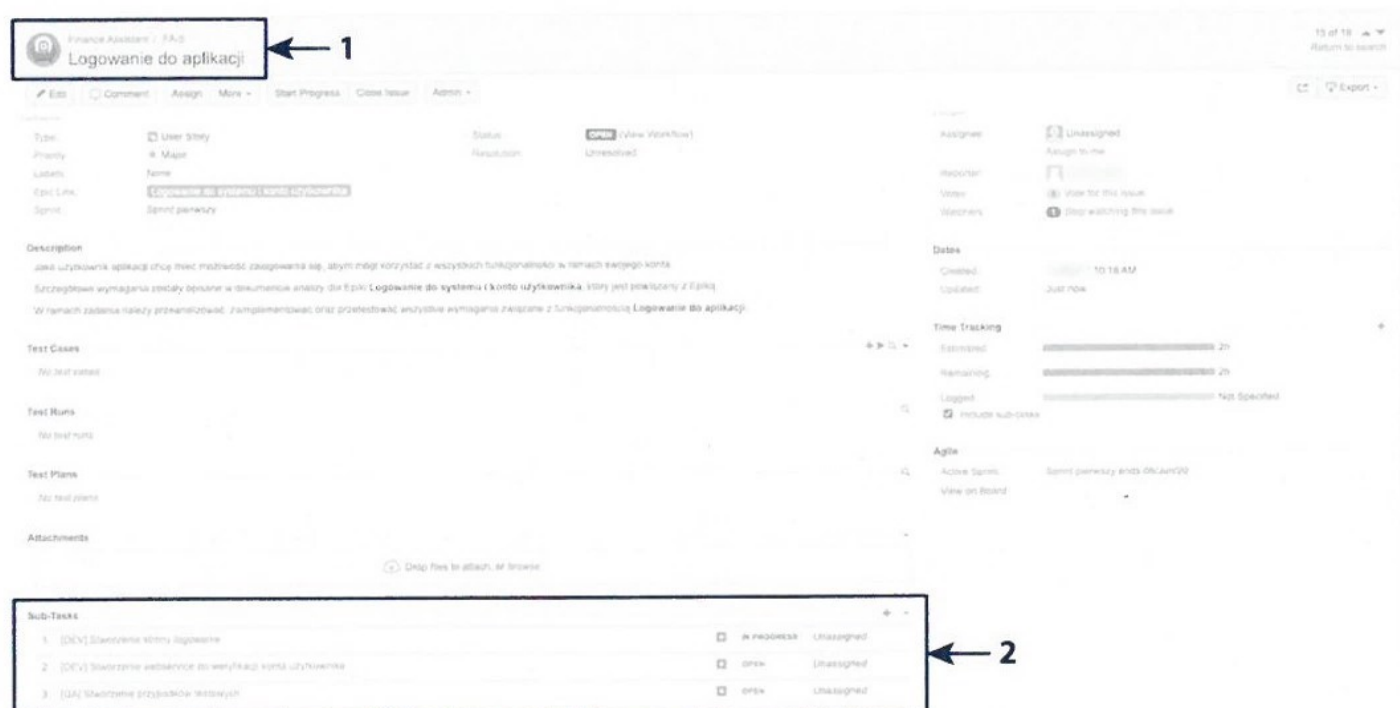
Tak wygląda ekran logowania dla Jira cloud (rysunek 1.4):



Rysunek 1.4. Ekran logowania w narzędziu Jira

Warto zaznaczyć, że ekran logowania do wersji cloud różni się wizualnie od wersji serwerowej. Jira dostosowuje się do projektu. Zależnie od metodyki, w jakiej projekt jest realizowany, narzędzie daje nam całą paletę różnorodnych funkcji. Poniżej (rysunek 1.5) zaprezentowano widok przykładowego zadania w systemie Jira wraz z podzadaniami (ang. *subtasks*). Można na nim zauważyć, że:

1. zadanie główne wyświetlane jest z nazwą projektu;
2. widoczne są wszystkie podzadania powiązane z zadaniem głównym.



Rysunek 1.5. Widok projektu w systemie Jira

W momencie gdy wszystkie podzadania wejdą w status świadczący o zakończonych pracach (może to być **Resolved** lub **Closed**), uznaje się, że zadanie główne może być wdrażane do środowiska (docelowego) produkcyjnego.

DEFINICJA

Środowisko produkcyjne — zainstalowane w siedzibie klienta, firmy tworzącej oprogramowanie lub w chmurze sprzęt i oprogramowanie, w których moduł (lub system) będzie używany. W skład oprogramowania mogą wchodzić systemy operacyjne, bazy danych i inne aplikacje.

Dodatkowym atutem narzędzia Jira jest definiowanie cyklu życia (ang. **workflow**) błędu.

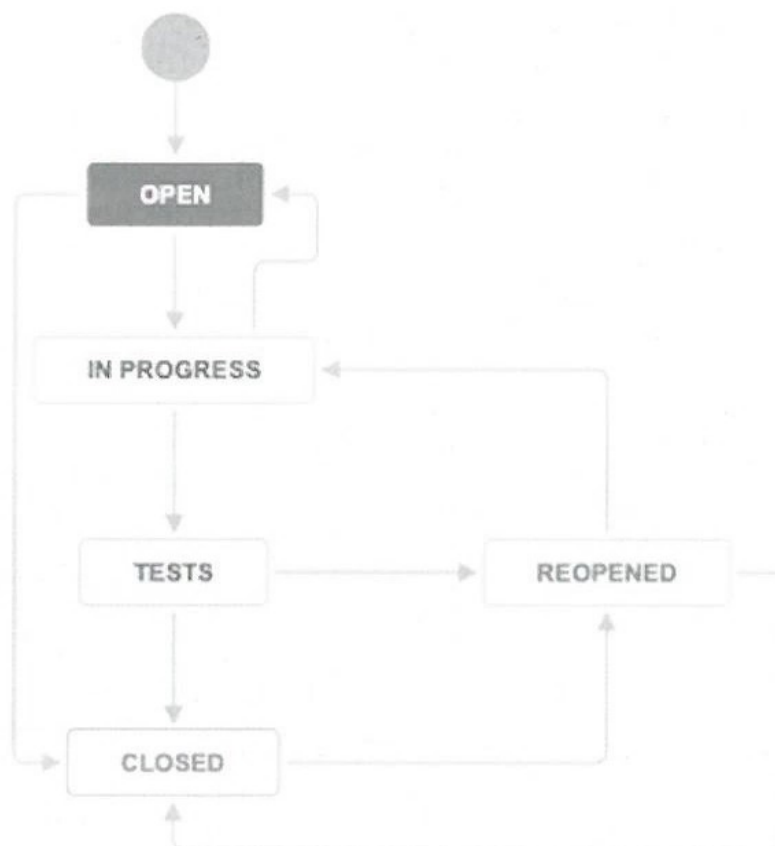
DEFINICJA

Cykl życia błędu to proces, przez który przechodzi usterka oprogramowania. Rozpoczyna się, gdy błąd zostanie wykryty, a kończy w chwili jego usunięcia, po wcześniejszym upewnieniu się, że nie został zduplikowany.

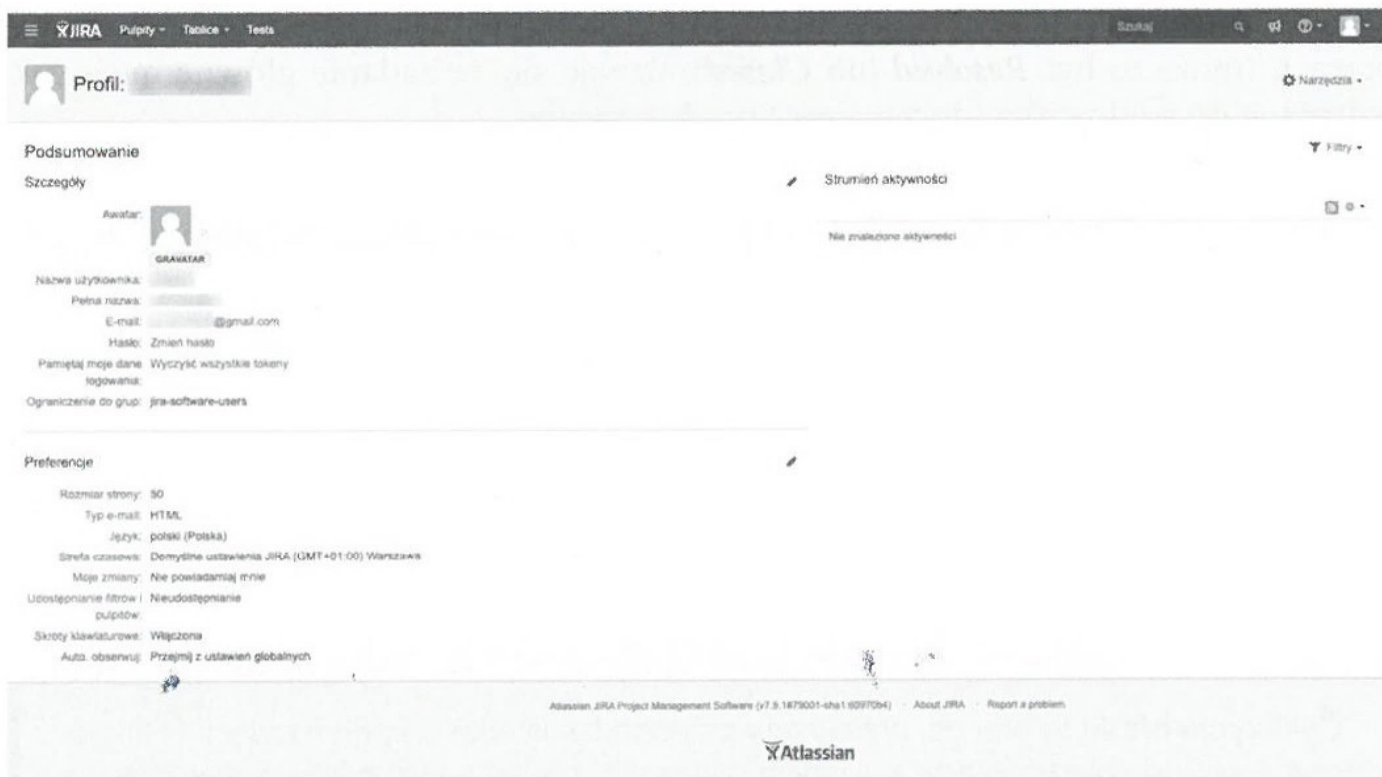
Na rysunku 1.6 przedstawiony jest diagram, który możemy dowolnie modyfikować. Każdy projekt może mieć odrębny **workflow**.

Rysunek 1.6.

Przykładowy cykl życia błędu (ang. workflow)



W **Panelu użytkownika** (rysunek 1.7) możemy zmienić preferencje co do wyświetlania stron oraz sprawdzić historię aktywności.



Rysunek 1.7. Konto użytkownika

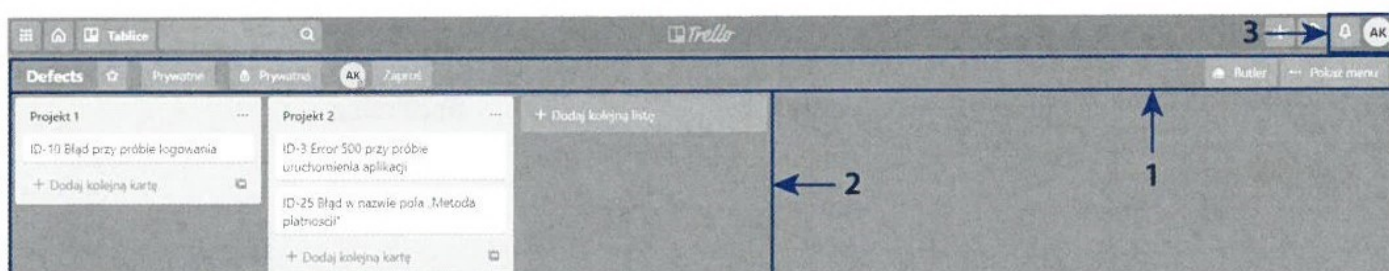
Dodatkowe informacje o narzędziu Jira zostaną podane w rozdziale 4.

1.1.3. Trello

Trello jest narzędziem bezpłatnym, które może być wykorzystywane zarówno w projektach komercyjnych, jak i w celach prywatnych (np. do planowania podróży, budżetu domowego, remontu).

Poniżej przedstawiono podgląd na widok tablicy Trello (rysunek 1.8). Można na nim zauważyć trzy grupy menu:

1. Menu operacji dotyczących udostępniania/prywatności tablicy.
2. Widok kart.
3. Konto użytkownika wraz z opcją powiadomień.



Rysunek 1.8. Widok na tablicę Trello

Dodatkowe informacje o Trello zostaną zaprezentowane w rozdziale 4.

W dalszej części podręcznika zostanie omówiony jeszcze TestLink — darmowe narzędzie typu *open source*.

1.1.4. Dobór narzędzi do zarządzania projektem

Poniżej przedstawiono krótkie zestawienie porównawcze narzędzi, które ułatwi Ci dopasowanie narzędzi do projektu. Zostały w nim uwzględnione ich kluczowe funkcje, dzięki którym wspierają prowadzenie projektu.

Tabela 1.1. Zestawienie porównawcze narzędzi

	HP ALM	Jira	Trello	TestLink
Płatna licencja	✓	✓		
Możliwość rozszerzania funkcji dzięki płatnym dodatkom	✓	✓	✓	
Personalizacja raportów	✓	✓		
Definiowanie cyklu życia błędu	✓	✓		
Wykonywanie przypadków testowych	✓			✓
Obsługa testów automatycznych	✓	✓		✓

1.2. Narzędzia wykorzystywane w wytwarzaniu oprogramowania

Jednym z elementów wytwarzania oprogramowania jest tworzenie aplikacji za pomocą wybranego języka programowania. W tym podręczniku skupiamy się na języku JavaScript. Zaczniemy od opisu elementów, na jakie należy zwrócić szczególną uwagę podczas instalacji oprogramowania.

Narzędzia pomocne do napisania aplikacji przeglądarkowej to Git i WebStorm.

1.2.1. Instalacja programów WebStorm i Git

1.2.1.1. Git — system kontroli wersji

Wyobraźmy sobie wspólne pisanie fragmentów kodu przez wielu użytkowników i przenoszenie tych kodów za każdym razem do chmury. Jak zapewnić, by zmiany dodawane przez dowolnego programistę były widoczne dla pozostałych współpracowników?

Gdy kolega z zespołu pobierze plik, a następnie coś w nim zmieni, wówczas, by było wiadomo, że kod został zmieniony, można dodać katalog z datą i godziną zmiany i to w nim właśnie przechowywana będzie nowa wersja pliku, podczas gdy oryginalna pozostanie niezmieniona.