

AzureDevOps

Mocne narzedzie dla zespolow programistycznych, ktore laczy w sobie system kontroli wersji (git), analize danych co do wydajnosci zespolu, metody zarzadzania projektami np. AGILE, status i zmiany w projekcie oraz duzo innych narzedzi.

1. Create new project

Klikamy New Project, wpisujemy nazwe oraz opis.

Advanced - Work item process:

Tu wybieramy metodologie wytwarzania oprogramowania, np. AGILE.

Advanced - Version control:

Tu wybieramy system kontroli wersji, domyslnie – git, ale rowniez istnieje mozliwosc wyboru Team Foundation Version Control.

Prywatny projekt jest gratis do 5 uzytkownikow.

2. Overview

a. Summary

Pierwsze co my widzimy to jest ogolny przeglad naszego projektu (Summary). Mozemy tu zobaczyc nasza statystyke projektu.

b. Dashboards

Narzędzie dla analizy pracy zespolu oraz jego wydajnosci.

c. Wiki

Rozdzial ktory mozna wykorzystywac do przechowywania jakis informacji o projekcie oraz dokumentacji.

Publish Code As Wiki – publikacja wybranego pliku z naszego repozytorium jako wiki;

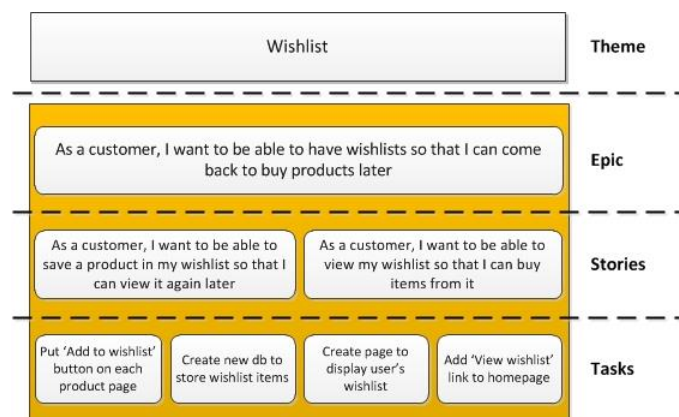
Create project wiki – tworzy samodzielnie wiki;

3. Boards

Implementacja zwinnych metodologii . W naszym przypadku implementacja AGILE'a.

a. Work items

Mam nastepujace zdjecie ktore opisuje Work Item'y:



Theme – to jest glowny temat naszej pracy , oraz zbior Epic'ow;

Epic - to jest powiedzenie ze strony uzytkownika o mozliwosciach i funkcjonalu produktu nad jakim my

pracujemy;

Stories – to Epic który jest rozbijany na części, też w imieniu użytkownika;

Tasks – poszczególne kroki dla implementacji zadanego funkcjonalu;

b. Boards

Standardowo 4 kolumny: New, Active, Resolved, Closed. Mamy elastyczny widok i możemy jego dopasować do naszych potrzeb. Na przykład dopasowując nazwy kolumn do naszych potrzeb. Zadania, które są stworzone w projekcie w łatwy sposób możemy przeciąć pomiędzy kolumnami, zmieniając ich status.

c. Backlogs

To jest wyświetlenie elementów pracy (work item'om) jako listę, w porównaniu do Boards, które wyświetla to jako tablice.

d. Sprints

Za pomocą Sprintów możemy tworzyć czasowe bloki zadań dla użytkowników. Sprints są definiowane dla projektu, a następnie przydzielone do poszczególnych członków zespołu lub małych zespołów.

Najczęściej czasowe ograniczenie na sprinty jest od 1 do 4 tygodni, pod czas tego czasu zespół przydzielony do sprintu ma swobodę zarządzać swoim czasem jak chcą. Głównie muszą zmieścić się w podanych przez administratora terminach.

e. Queries

Zapytania to kombinacja kilku „filtrow” które są stosowane do naszych itemów. Jest używana do filtracji naszych roboczych itemów, w przypadku jeżeli mamy kilka projektów z tysiącami zadań lub dużą ilością użytkowników.

4. Repos

a. Files

Tu najpierw musimy albo zaimportować, albo sklonować repozytorium. Pokażę kilka rzeczy: importujemy repozytorium mojego projektu z programowania zespołowego; klonujemy nasz repozytorium z DevOps na lokalną maszynę przez Visual Studio; dodamy plik w lokalnym repo, komitujemy go i robimy push'a; zarządzanie zadaniami z poziomu Visual Studio;

b. Commits

Mamy graf oraz możemy zobaczyć informacje o commit'ach: ich numer, nazwę autora, zmiany które on prowadzi i t.d.

c. Pushes

Zwykła lista zrobionych push'ów, możemy ich filtrować.

d. Branches

Gałęzi naszego projektu, na razie mamy 3 i za chwylę spróbujemy ich połączyć, czyli zrobić Pull'a

e. Tags

Narzędzie przydatne dla szybkiego dostępu do jakichś danych, np. zrobimy tag do jakiegoś commit'a. Czyli to działa jako wskaźnik do podanych danych.

f. Pull Requests

Pull Requests łączą przegląd i scalenie kodu w jeden proces współpracy. Gdy skończymy naprawiać jakiś błąd lub nową funkcję w gałęzi, to tworzymy nowy Pull Request. Możemy dodać członków zespołu do

Pull Request'u aby mogli przejrzeć nasze zmiany i dyskutować o tym, na przykład przed scaleniem kodu z gałęzią główną.

5. **Pipelines**

Hostowana w chmurze ciągła integracja i ciągłe dostarczanie(CI/CD). Daje możliwość budować, testować i rozmieszczać aplikacje w Azure Chmurze i innych platformach .

6. **Test plans**

Narzędzie do testowania kodu na Azure. Ma 3 główne typy testowania : test plans , test suites , test cases . Są one przechowywane w repozytorium pracy jako specjalne typy elementów pracy, umożliwiając eksportowanie i udostępnianie ich zespołowi oraz korzystanie z bliskiej integracji wszystkich zadań DevOps.

7. **Artifacts**

Rozszerzenie usług Azure DevOps Services i Azure DevOps Server. Jest przydatne dla wspólnego wykorzystania pakietów Maven , Nuget z ogólnodostępnych i zamkniętych źródeł . Pozwala tworzyć , rozmieszczać i wykorzystywać pakiety wspólnie z zespołem . Można integrować z Azure Pipelines .