

[← GitLab](#)

# GitLab w rozwoju oprogramowania

## – część 3: ulepszenia i wskazówki

**GITLAB**

21 maja 2019 · 16 min czytania

**Armin Orlik**[Wstęp](#)[Wprowadzaj ulepszenia:](#)[Wskazówki](#)[Scenariusz użytkowania GitLab](#)[Workflow](#)[Podsumowanie](#)[Wstęp](#)[Wprowadzaj ulepszenia:](#)[Wskazówki](#)[Scenariusz użytkowania GitLab](#)[Workflow](#)[Podsumowanie](#)

## Wprowadzaj ulepszenia:

### Szablony zgłoszeń i MR-ów

[Szablony zgłoszeń i MR-ów](#) pozwolą Ci stworzyć w projekcie kontekstowe wzory pól opisu dla zgłoszeń i merge requestów.

Aby to uczynić możemy skorzystać z [Markdowna](#), a następnie dodać do domyślnego brancha repozytorium. W następnej kolejności możemy uzyskać do nich dostęp przy pomocy listy rozwijanej, w trakcie tworzenia zgłoszenia lub MR-a.

To rozwiązanie pozwala oszczędzać czas spędzony nad opisywaniem zgłoszeń i MR-ów, jak i również ujednolicić opisy tak, aby łatwiej było je śledzić.

Ponieważ mamy możliwość tworzenia wielu szablonów, mogą one służyć konkretnym celom. Na przykład, jeden może obsługiwać propozycje nowych funkcji, a inny raporty o błędach. Przykłady użycia prezentujemy w [projekcie GitLab CE](#). Dostęp do źródeł możliwy jest tylko po zalogowaniu się do systemu GitLab.

## Milestone'y

[Milestone'y](#) to narzędzie oferowane przez GitLab, które służy do śledzenia postępów zespołu przy pracy nad wspólnym celem w określonym czasie.

Celem może być cokolwiek, co wymaga zaangażowania i pracy zespołowej w określonym terminie. Może to być np. nowe wydanie, uruchomienie produktu, ukończenie określonych zadań, czy też stworzenie grupy projektów do ukończenia w danym kwartale.

Możemy na przykład utworzyć milestone na pierwszy kwartał danego roku i przypisać do niego wszystkie zgłoszenia i MR-y przeznaczone do ukończenia w tym terminie. Posłużmy się prostym przykładem: nasza firma jest organizatorem wydarzenia. Proces ten składa się w wielu czynności i decyzji. Aby uzyskać pełen obraz postępu w przygotowaniach tworzymy milestone'a, który pozwala nam łatwo i szybko prześledzić wszystkie elementy procesu. Zarówno te, które zostały już zrealizowane, jak i oczekujące na podjęcie działań.

## Wskazówki

## Dla zgłoszeń i MR-ów

Gdy tworzysz opis zgłoszenia lub MR-a:

- wpisz #, aby rozwinąć listę zarejestrowanych zgłoszeń,
- wpisz !, aby rozwinąć listę zarejestrowanych MR-ów,
- wpisz /, aby uruchomić komendy slash,
- wpisz :, aby dodać emotikonki (również dla komentarzy w linii).

Do komentarzy w linii możesz dołączyć pliki graficzne (jpg, png, gif) lub video, używając przycisku **Dodaj plik**.

[Automatycznie dodawaj oznaczenia](#) przy pomocy [GitLab Webhooks](#).

[Nowy sposób dodawania cytatów](#): użyj >>> aby rozpocząć i zamknąć cytat.

Nowy sposób tworzenia [list zadań](#):

## Subskrypcje

Co w przypadku jeśli znajdziemy zgłoszenie lub MR-a, którego chcemy śledzić? Rozwijamy menu po prawej stronie i klikamy [Subscribe](#), aby dostawać powiadomienia, gdy ktoś doda nowy komentarz. Jeśli chcemy śledzić kilka zgłoszeń i MR-ów naraz, używamy [subskrypcji grupowej](#).

## Dodawanie poleceń (TO-DO)

Jeśli planujemy kiedyś podjąć pracę nad śledzonym zgłoszeniem lub MR-em lub po prostu chcemy dodać go do swojej listy zadań (TO-DO), rozwijamy panel z prawej strony i [klikamy Add todo](#).

## Wyszukiwanie zgłoszeń i MR-ów

GitLab udostępnia nam proste narzędzie do wyszukiwania zgłoszeń lub MR-ów, w których dotychczas braliśmy udział. Aby móc z niego skorzystać rozwijamy panel po lewej stronie i wybieramy Issues lub Merge Requests, żeby zobaczyć tylko te elementy, które są do nas przypisane. Z tego miejsca lub z dowolnego issue trackera, możemy filtrować zgłoszenia lub MR-y po autorze, osobie przypisanej, milestone, etykiecie i wadze. Możemy także szukać zgłoszeń otwartych, zmergowanych, zamkniętych albo wszystkich równocześnie.

## Przenoszenie zgłoszeń

Organizacja zgłoszeń jest prosta – jeśli zgłoszenie znajdzie się w niewłaściwym projekcie, wybieramy Edit, a następnie [przenosimy je](#) do prawidłowego miejsca.

## Snippety

Jeśli w naszym projekcie często korzystamy z tego samego fragmentu kodu lub szablonu, możemy utworzyć snippet, który bardzo ułatwi nam pracę. Aby to zrobić musimy rozwinąć panel po lewej stronie, i kliknąć w **Snippets**. Wszystkie nasze snippety będą przechowywane dokładnie w tym miejscu. Możemy również zmieniać ich ustawienia prywatności (publiczny, wewnętrzny – tylko dla zalogowanych użytkowników GitLab oraz prywatny).

## Scenariusz użytkowania GitLab Workflow

Na koniec zaprezentujemy przykładowy scenariusz workflow w oparciu o funkcjonalności i środowisku GitLab.

Przypuśćmy, że pracujemy w firmie tworzącej oprogramowanie. Stworzyliśmy właśnie nowe zgłoszenie dotyczące nowej funkcjonalności, która ma być wdrożona w jednej z naszych aplikacji.

## Strategia etykietowania

Dla aplikacji, o której mowa, mamy już oznaczenia takie jak “dyskusja”, “backend”, “frontend”, “w trakcie pracy”, “pre-produkcja”, “gotowe”, “dokumentacja”, “marketing” i “produkcja”. Każde z nich ma już własną listę na Tablicy Zgłoszeń (Issue Board). Do naszego zgłoszenia przypisane jest obecnie oznaczenie “dyskusja”.

Gdy dyskusja w issue trackerze zakończy się porozumieniem, nasz zespół backend zaczyna pracę nad zgłoszeniem, więc ich lider przenosi je z listy “dyskusja” na “backend”. Pierwszy deweloper, który zaczął pisać kod i przypisał siebie do zgłoszenia, dodaje oznaczenie “w trakcie pracy”.

## Kodowanie i commit

Ten sam deweloper odnosi się do numeru zgłoszenia w pierwszej wiadomości commitu. Po skończeniu pracy wykonuje push do funkcjonalnego brancha i tworzy merge requesta, dołączając wzór zamknięcia zgłoszenia do jego opisu. Jego zespół przegląda kod i sprawdza czy wszystkie testy i buildy działają poprawnie.

## Używanie tablicy zgłoszeń (Issue Board)

Po zakończeniu pracy, zespół backend usuwa znacznik “w trakcie pracy” i przenosi zgłoszenie z listy “backend” do “frontend” na tablicy zgłoszeń. W ten sposób zespół frontend’u dowiaduje się, że może już rozpocząć pracę.

## Wdrażanie pre-produkcji

Gdy front-endowiec rozpocznie pracę nad zgłoszeniem, ponownie dodaje znacznik “w trakcie pracy” i przypisuje się do zgłoszenia. Gdy skończy pracę, wystawia swoje rozwiązanie na środowisku **pre-produkcyjnym**. Znacznik “w trakcie pracy” zostaje usunięty, a zgłoszenie przeniesione na listę “pre-produkcji” na tablicy zgłoszeń.

## Praca zespołowa

Na koniec, gdy rozwiązanie działa już poprawnie, zespół deweloperski przenosi je na listę “gotowe”.

Nadszedł czas, aby grupa technical writerów udokumentowała nową funkcjonalność. Zgodnie z dotychczasową praktyką, przypisany tech writer dodaje znacznik “dokumentacja”. Zespół marketingowy zaczyna równocześnie pracę nad kampanią reklamową, więc członek zespołu dodaje znacznik “marketing”. Gdy tech writer ukończy pracę nad dokumentacją, usuwa swój znacznik. Gdy zespół marketingowy ukończy zadanie, przesuwa zgłoszenie z listy “marketing” na listę oznaczoną jako “produkcja”.

## Wdrożenie do produkcji

Osoba odpowiedzialna za nowe wydanie merguje MR-a i wprowadza nową funkcjonalność na środowisko **produkcyjne**, a zgłoszenie zostaje **zamknięte**.

## Podsumowanie

GitLab Workflow pomaga zespołom deweloperskim tworzyć oprogramowanie przy użyciu pojedynczej platformy. W swoim działaniu jest, między innymi:

**Efektywny** – dostarcza wiele opcji co pomaga osiągać założone cele.

**Wydajny** – pozwala maksymalnie zwiększyć produktywność minimalnym nakładem sił i środków.

**Produktywny** – zapewnia możliwość sprawnego planowania i wydajnej pracy.

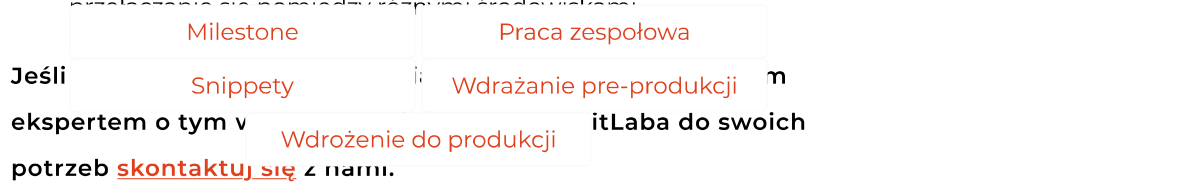
**Łatwy w użyciu** – swoją funkcjonalnością zastępuje wiele różnych narzędzi, niezbędnych w procesie developmentu.

Dostosowujemy GitLaba indywidualnie do potrzeb i to wszystko.

Jedna konfiguracja, a wiele możliwości.

**Szybki** – działa w oparciu o jedną platformę, nie tracimy czasu na

przemieszczanie się pomiędzy różnymi środowiskami:



## Armin Orlik

Inżynier i konsultant w obszarach DevOps, Automation, Cloud, konteneryzacja. Bogate doświadczenie we wdrażaniu i utrzymaniu praktyk DevOps. Obecnie pasjonat rozwiązań chmurowych w szczególności bazując na AWS. Interesuje się szeroko pojętą automatyzacją oraz projektowaniem mikroservisów i aplikacji serverless oprogramowania.

### Technology-Driven Results

#### Kontakt

Siedziba Deviniti  
ul. Sudecka 153  
53-128 Wrocław

- - - - ▶

#### Nagrody

#### Nasza Oferta

[Custom Development](#)

[Mobile Development](#)

[Atlassian Services](#)

[Atlassian Apps](#)

[Cloud Services](#)

#### Firma

[O nas](#)

[Blog](#)

[Kontakt](#)

[Kariera](#)

[Partnerzy](#)

[Wsparcie](#)



Cieszymy się, że dotarłeś aż tutaj! To znak, że jesteś wytrwałym astronautą. Tak samo, jak my! Od 2004 roku misją Deviniti jest pomaganie biznesom w przejściu przez cyfrową transformację. Doradzamy w zakresie najlepszych narzędzi, które wystrzelą efektywność procesów biznesowych. Towarzyszymy Ci podczas każdego kroku rozwoju aplikacji: od konsultacji i analizy biznesowej, przez wdrożenie, po utrzymanie i wsparcie techniczne. Skontaktuj się z nami i odkryj Wszechświat Transformacji Cyfrowej!

© [Deviniti](#) 2020

[Polityka Prywatności & Cookies](#)