

O'REILLY®



GitHub

Przyjazny przewodnik

POZNAJ MOŻLIWOŚCI GITA I GITHUBA!

Tytuł oryginału: Introducing GitHub

Tłumaczenie: Piotr Cieślak

ISBN: 978-83-283-0503-8

© 2015 Helion S.A.

Authorized Polish translation of the English edition of Introducing GitHub,
ISBN 9781491949740 © 2015 Pragmatic Learning, Inc.

This translation is published and sold by permission of O'Reilly Media, Inc.,
which owns or controls all rights to publish and sell the same.

All rights reserved. No part of this book may be reproduced or transmitted in any form
or by any means, electronic or mechanical, including photocopying, recording or by any
information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu
niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą
kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym,
magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi
bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce
informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za
ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub
autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności
za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/github>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

- Kup książkę
- Poleć książkę
- Oceń książkę

- Księgarnia internetowa
- Lubię to! » Nasza społeczność

Spis treści

Przedmowa	7
Do kogo jest adresowana ta książka?	7
Nie tylko kod	8
Komu nie przyda się ta książka?	8
Jak korzystać z niniejszej książki?	9
Konwencje zastosowane w książce	9
Podziękowania	9
1. Wstęp	11
Czym jest Git?	11
Czym jest GitHub?	11
Dlaczego warto używać Gita?	12
Dlaczego warto używać GitHuba?	13
Najważniejsze pojęcia	13
2. Przeglądanie	17
Zapoznanie się ze stroną projektu	17
Wyświetlanie pliku README.md	19
Przeglądanie historii zmian	20
Wyświetlanie zgłoszeń	20
Wyświetlanie problemów	22
Wyświetlanie pulsu	23
Wyświetlanie wykresów w GitHubie	25
Wykres współpracowników	25
Wykres zmian	27
Wykres częstotliwości zmian w kodzie	28
Wykres „karta perforowana”	29
Wykres sieciowy	30
Lista uczestników	30
Wykres ruchu sieciowego	32

3. Edycja	33
Udział w projekcie za pośrednictwem odgałęzień	33
Dodawanie pliku	34
Tworzenie zgłoszenia	36
Edytowanie pliku	44
Zmiana nazwy albo przenoszenie pliku	47
Praca z folderami	48
Tworzenie folderu	49
Zmiana nazwy folderu	49
Ograniczenia edytowania w serwisie GitHub	49
4. Współpraca	51
Wprowadzanie zmian w odgałęzieniu	51
Tworzenie zgłoszenia z poziomu gałęzi	54
Współpraca przy zgłoszeniach	56
Zaangażowanie konkretnych osób w zgłoszenie	56
Przeglądanie zgłoszeń	57
Komentowanie zgłoszeń	57
Uatrakcyjnianie komentarzy	58
Modyfikowanie zgłoszeń	59
Testowanie zgłoszenia	61
Akceptowanie zgłoszenia	62
Kto powinien akceptować zgłoszenia?	63
Powiadomienia o zgłoszeniach	63
Zalecenia dotyczące zgłoszeń	64
Problemy	65
Definiowanie nowego problemu	65
Zarządzanie kamieniami milowymi w ramach problemów	67
Zarządzanie etykietami w ramach problemów	68
Komentowanie problemów	70
Odwoływanie się do problemów w zmianie	70
Zalecenia dotyczące problemów	70
Wiki	71
Rozpoczęcie pracy z wiki	72
Dodawanie stron do wiki i odwoływanie się do nich	74
Informacyjne strony WWW w GitHubie	75
Tworzenie strony internetowej dla projektu	76
Tworzenie strony internetowej dla siebie albo dla organizacji	78

5. Tworzenie i konfigurowanie	81
Tworzenie repozytorium	81
Dodawanie współpracowników	84
Konfigurowanie repozytorium	86
Integracja z innymi systemami	88
Repozytoria osobiste i repozytoria organizacji	95
Tworzenie organizacji	95
Zarządzanie zespołami	96
6. Pobieranie	101
Po co klonować repozytorium?	101
GitHub w wersji dla systemu Mac OS	102
Przesyłanie zmian przy użyciu programu GitHub dla Mac OS-a	110
Wyświetlanie zmian w programie GitHub dla Mac OS-a	112
Program GitHub w wersji dla systemu Windows	114
Przesyłanie zmian przy użyciu programu	
GitHub dla Windows	121
Konfigurowanie narzędzi do obsługi Gita	
z poziomu wiersza poleceń systemu Windows	123
7. Kolejne kroki	125
Skorowidz	127

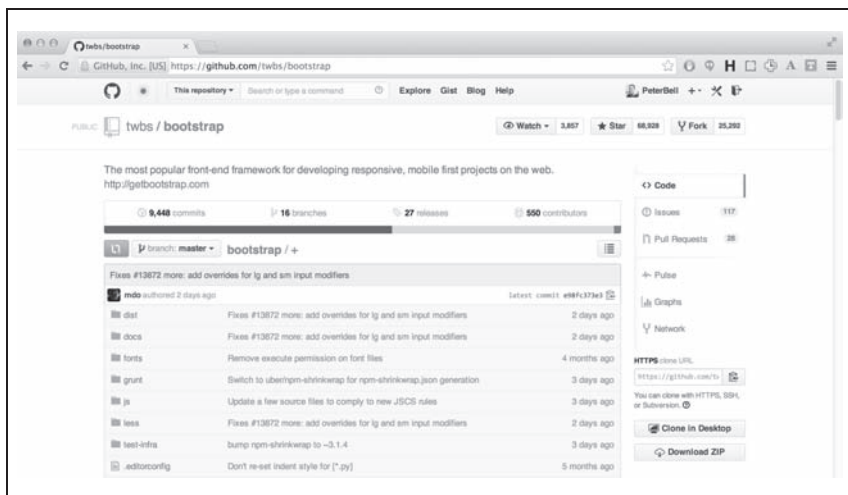
Przeglądanie

W tym rozdziale zapoznasz się z możliwościami przeglądania bieżącego stanu projektu, co pozwoli Ci zobaczyć, co się w nim dzieje. Jako przykładu użyjemy Bootstrapa (<http://getbootstrap.com/>) — popularnego projektu open source.

Zapoznanie się ze stroną projektu

Bootstrap to narzędzie umożliwiające szybkie projektowanie atrakcyjnie wyglądających stron internetowych. Odwiedź stronę projektu w serwisie GitHub (<https://github.com/twbs/bootstrap>). Na stronie głównej znajduje się wiele informacji — zacznijmy od przejrzania najważniejszych jej elementów (rysunek 2.1).

Jedną z pierwszych rzeczy, na którą warto zwrócić uwagę, jest widoczna w lewym górnym rogu nazwa projektu („bootstrap”) oraz informacja o tym, że jego właścicielem jest użytkownik (w tym wypadku konkretnie organizacja) o nazwie „twbs”. Jeśli odwiedziłbyś stronę <https://github.com/twbs>, to mógłbyś przejrzeć wszystkie projekty tej organizacji dostępne w GitHubie. Po lewej stronie nazwy organizacji znajduje się ikona oznaczająca, że jest to repozytorium publiczne, a więc takie, które każdy może przeglądać. Wielu projektom, nad którymi będziesz pracować, może towarzyszyć symbol zamkniętej kłódki, oznaczający, że są to projekty prywatne i mogą być oglądane tylko przez osoby będące współpracownikami w projekcie.



Rysunek 2.1. Strona główna repozytorium Bootstrapa w serwisie GitHub

Jak można zobaczyć po prawej stronie nazwy projektu, w chwili robienia zrzutu ekranu pokazanego na rysunku 2.1 repozytorium Bootstrapa *obserwowało* (Watch) 3857 osób, aby otrzymywać informacje za każdym razem, gdy w projekcie zostaną wprowadzone jakieś zmiany. Ponadto 68 928 osób oznaczyło projekt *gwiazdką* (Star), aby w ten sposób dodać go do swoich ulubionych, a 25 292 osoby zrobiły *odgałęzienie* całego repozytorium (Fork), czyli utworzyły w serwisie GitHub jego kopię, w której mogą wprowadzać zmiany i udostępniać je innym użytkownikom.

Nieco niżej na stronie znajduje się krótki opis projektu, a pod nim widnieje informacja, że w projekcie wprowadzono łącznie 9448 zmian (*commit*), aktualnie jest rozwijanych 16 gałęzi historii (*branch*), 27 wydań zostało oznaczonych jako zalecane (tzw. wersje produkcyjne, czyli *release*), a nad kodem pracowało w sumie 550 osób (*contributor*).

Jak widać, aktualnie przeglądamy gałąź główną projektu, czyli jesteśmy w folderze głównym repozytorium o nazwie *bootstrap*, a ostatnia zmiana w tej gałęzi ma następujący opis: „Fixes #13872 more: add overrides for lg and sm input modifiers” (cokolwiek by to oznaczało). Zmiany tej dokonał użytkownik GitHuba o pseudonimie „mdo” (<https://github.com/mdo>). Jeszcze niżej na rysunku znajdują się foldery (zwane też *katalogami*) oraz pliki umieszczone w folderze głównym projektu (tzw. *root*).

Wyświetlanie pliku README.md

Jeśli w folderze *root* projektu znajduje się plik o nazwie *README.md*, to jego zawartość zostanie wyświetlona pod listą folderów i plików na stronie głównej tego projektu. Wspomniany plik zawiera ogólne omówienie projektu oraz pomocnicze informacje, które mogą się przydać jego współautorom, takie jak sposób instalacji oprogramowania albo uruchomienia testów, dopuszczalne zastosowanie kodu i możliwe rodzaje współpracy.

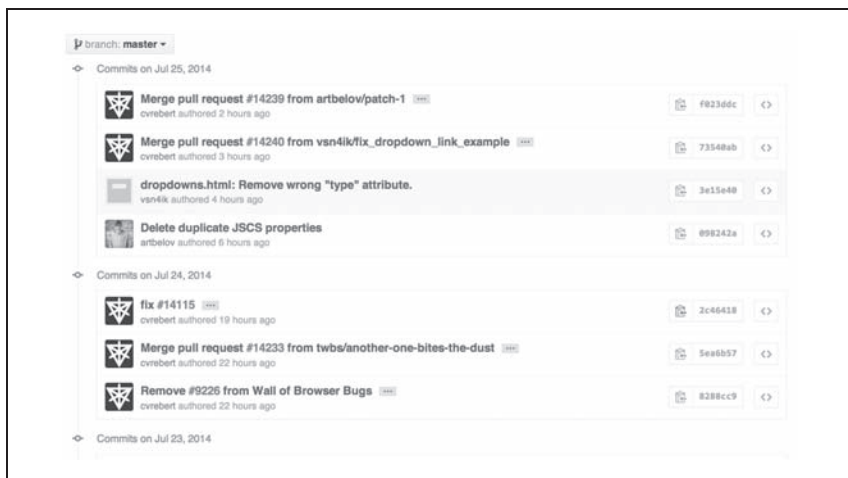
Obecnie w plikach README coraz częściej znajdują się infografiki ilustrujące aktualny stan prac, generowane na przykład przez automatyczne narzędzia do testowania. Jak pokazano na rysunku 2.2, na stronie projektu Bootstrap znajdują się numery wersji dwóch innych projektów, od których Bootstrap jest zależny. Widać na nim też, że automatyczne testy zostały wykonane bez przeszkód, a zależności są aktualne; ponadto na infografice widnieją numery wersji przeglądarek i systemów operacyjnych, w których Bootstrap powinien poprawnie funkcjonować.



Rysunek 2.2. Zawartość pliku README.md projektu Bootstrap

Przeglądanie historii zmian

Historia zmian jest znakomitym źródłem informacji na temat prac wykonanych w danej gałęzi projektu. Otwórz stronę projektu Bootstrap w serwisie GitHub (<https://github.com/twbs/bootstrap>) i kliknij odsyłacz z liczbą zmian — w chwili, gdy piszę te słowa, miał on postać *9,448 commits*, ale gdy Ty odwiedzisz wspomnianą stronę, liczba ta będzie oczywiście inna. Na ekranie wyświetli się lista zmian, począwszy od najnowszych (rysunek 2.3).

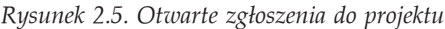


Rysunek 2.3. Lista najnowszych zmian w projekcie

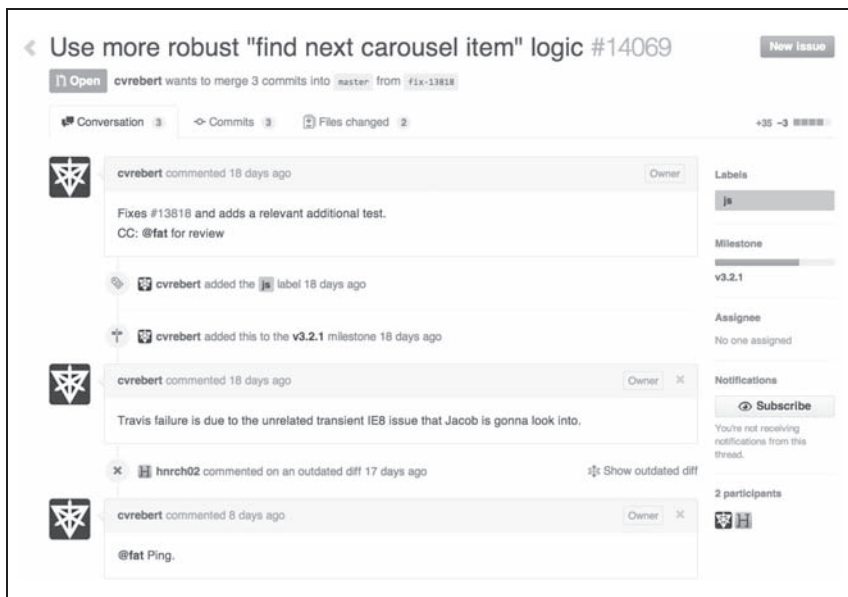
Kliknięcie dowolnej zmiany spowoduje wyświetlenie powiązanego z nią komunikatu, który powinien wyjaśniać *przyczynę* jej wprowadzenia. Poniżej zostaną wyświetlone pliki, które dodano, usunięto lub zmodyfikowano w ramach danej zmiany, przy czym fragmenty usunięte zostaną wyróżnione kolorem czerwonym, dodane zaś — zielonym (rysunek 2.4).

Wyświetlanie zgłoszeń

Zgłoszenia (*pull requests*) pozwalają zorientować się w bieżącym stanie prac. Wróć na stronę główną i kliknij odsyłacz *Pull Requests*, znajdujący się po prawej stronie, u góry. W oknie przeglądarki pojawi się wówczas lista otwartych zgłoszeń. Są to na ogół funkcje lub poprawki, nad którymi aktualnie pracują uczestnicy projektu (rysunek 2.5).



Po kliknięciu dowolnego zgłoszenia wyświetlony zostanie jego tytuł wraz z krótkim opisem. Opisowi mogą towarzyszyć numery proponowanych zmian oraz komentarze użytkowników dyskutujących nad ich zaimplementowaniem (rysunek 2.6).



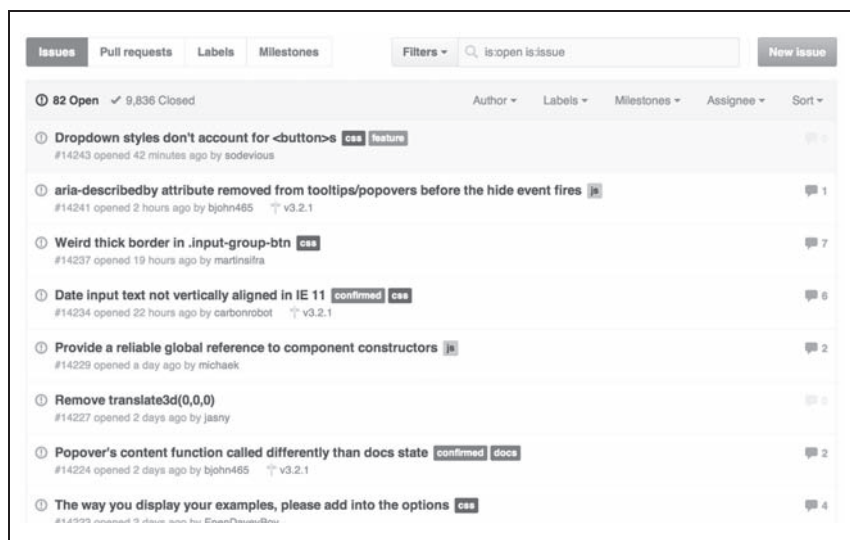
Rysunek 2.6. Jedno z niedawnych zgłoszeń

Przeglądanie zgłoszeń jest znakomitą sposobem na zapoznanie się z najnowszymi pracami toczącymi się w ramach projektu oraz aktualnym stanem zmian — niezależnie od tego, czy są to poprawki błędów, czy proponowane funkcje.

Wyświetlanie problemów

O ile zgłoszenia pozwalają określić aktualny stan prac nad poprawkami błędów i funkcjami, o tyle przeglądanie problemów (*issues*) umożliwia spojrzenie z szerszej perspektywy na te aspekty projektu, które wciąż oczekują na realizację. Zgłoszenia są często powiązane z problemami, ale w wielu projektach istnieją takie problemy, nad którymi nikt jeszcze nie zaczął pracować, a co za tym idzie — nie zgłosił ich.

Po kliknięciu odsyłacza *Issues* umożliwiającego wyświetlenie listy problemów domyślnie pokazuje się zestawienie tych z nich, które nadal są otwarte (rysunek 2.7).



Rysunek 2.7. Otwarte problemy dla danego projektu

Po kliknięciu wybranego problemu wyświetli się jego tytuł oraz dotyczące go komentarze. Jeśli w kwestii tego problemu zostały wprowadzone jakieś zmiany i wysłano je do GitHuba, a w komentarzu do tych zmian jest wzmianka o danym problemie, to zostaną one wyświetlone na stronie problemu, abyś mógł sprawdzić, co zostało zrobione. Jak pokazano na rysunku 2.8, jeden z użytkowników ma kłopoty z pewną funkcją Bootstrapa.

Wyświetlanie pulsu

Puls (odsylacz *Pulse*) to świetny sposób na oszacowanie bieżącej aktywności projektu. Puls można skonfigurować tak, by ilustrował działania z ostatniego dnia, ostatnich trzech dni lub ostatniego tygodnia albo miesiąca. Na rysunku 2.9 pokazano działania obejmujące tydzień.

Puls rozpoczyna się informacją o liczbie zgłoszeń włączonych do projektu, czyli scalonych (*merged*), oraz zgłoszeń proponowanych (*proposed*). Pokazuje też liczbę zgłoszeń zamkniętych i otwartych. Warto zapamiętać, że podana w pulsie liczba aktywnych zgłoszeń i zmian nie odnosi się do zgłoszeń

i zmian nierozpatrzonych, ale tylko do tych, które zostały zapoczątkowane i ukończone w wybranym okresie. Na przykład w chwili, gdy piszę te słowa, w projekcie Bootstrap było 15 scalonych i 8 proponowanych zgłoszeń, co daje łącznie 23 aktywne zgłoszenia w danym tygodniu, jednakże sumaryczna liczba otwartych zgłoszeń wynosiła 28.

Akapit znajdujący się poniżej stanowi zwięzłe podsumowanie ostatnich modyfikacji i zawiera dane o liczbie ich autorów, zmian w gałęzi głównej, całkowitej liczbie zmian we wszystkich gałęziach oraz liczbie plików dodanych, usuniętych lub zmodyfikowanych w obrębie gałęzi głównej. Oprócz tego znajduje się w nim informacja o liczbie linii tekstu, które zostały dodane lub usunięte; z tym zastrzeżeniem, że jeśli w pliku tekstowym zostanie zmieniona zawartość jednej linii, Git potraktuje to tak, jakby jedna linia została usunięta, a inna dodana na jej miejsce.

Po prawej stronie znajduje się wykres słupkowy ilustrujący zestawienie uczestników projektu, którzy w danym okresie dokonali największej liczby zmian. Pod nim widnieje lista tytułów scalonych i proponowanych zgłoszeń, po niej zaś lista zamkniętych i otwartych problemów. Podgląd pulsu kończy się listą „niezamkniętych dyskusji” (*unresolved conversations*), czyli zestawieniem wszystkich problemów i zgłoszeń, które zostały w jakiś sposób skomentowane, ale są nadal nierozwiązane.

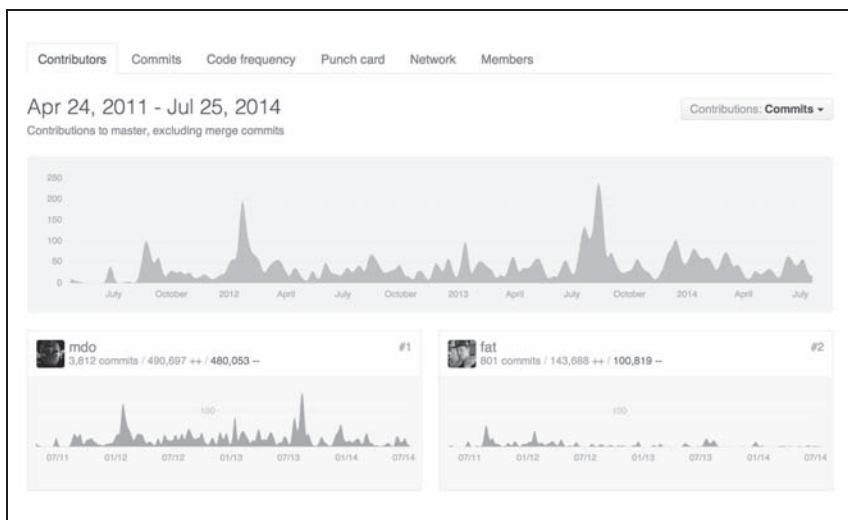
Wyświetlanie wykresów w GitHubie

O ile puls stanowi podsumowanie bieżącej aktywności, o tyle strony z wykresami (odsylacz *Graphs*) umożliwiają oszacowanie prac, jakie zostały wykonane w ramach danego projektu przez dłuższy czas.

Wykres współpracowników

Wykres współpracowników, pokazany na rysunku 2.10, przedstawia fluktuację zaangażowania w projekt w czasie na podstawie liczby operacji zmieniania, dodawania i usuwania. Główny wykres obejmuje wszystkie takie działania, pod nim zaś znajdują się mniejsze wykresy, ilustrujące aktywność poszczególnych uczestników projektu — od najbardziej do najmniej aktywnych.

Domyślny wykres zmian ilustruje liczbę zmian w gałęzi głównej w kontekście czasu, przy czym są to jedynie te zmiany, które zostały scalone w tej gałęzi. Jeśli ktoś z Twojego zespołu przez cały tydzień pracował nad gałęzią

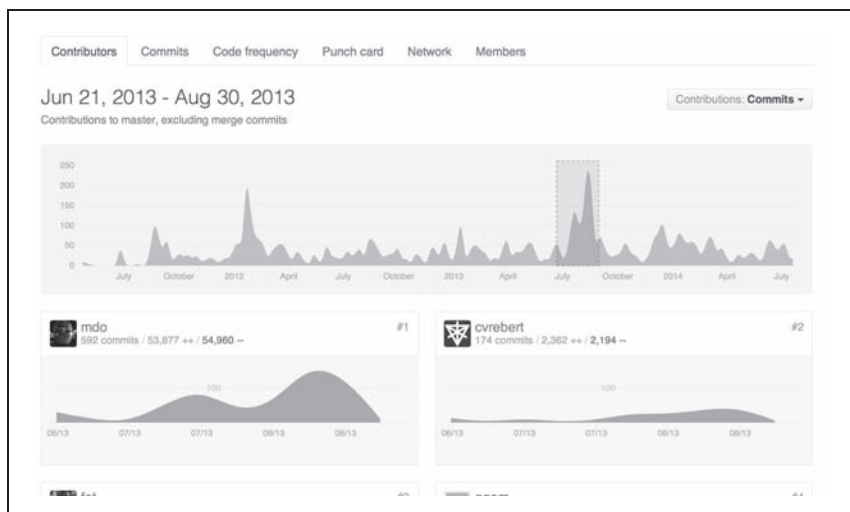


Rysunek 2.10. Wykres współpracowników projektu Bootstrap

poświęconą jakiejś funkcji, a jego praca nie została jeszcze włączona do gałęzi głównej, to aktywność tego członka zespołu nie zostanie uwzględniona aż do chwili, gdy funkcja uzyska status produkcyjny i zostanie scalona z gałęzią główną projektu.

Domyślnie wykresy obejmują cały czas funkcjonowania projektu. Jeśli chciałbyś wybrać węższy zakres czasu, kliknij wykres tam, gdzie zakres ten ma się zaczynać, a potem przeciągnij kursorem myszy i zwolnij jej przycisk w miejscu, w którym ma się on kończyć. Rysunek 2.11 przedstawia efekt takiej operacji, mającej na celu wyświetlenie zmian przeprowadzonych latem 2013 roku. Jak widać, główny diagram w górnej części strony pozostał bez zmian, z tym że w jego lewym górnym rogu widnieje informacja o wybranym zakresie czasu (od 21 czerwca do 30 sierpnia). Za to wykresy ilustrujące zmiany dokonane przez poszczególnych uczestników przedstawiają rozkład tych zmian tylko w wybranym okresie.

Nie istnieją wytyczne co do wielkości zmiany. Przyjmuje się regułę, że jeśli uczestnicy projektu nie zajmują się analizą problemów lub przeprowadzaniem testów, lecz piszą kod, to powinni wysyłać zmiany co 5 – 10 minut. Jednak w zależności od zespołu, z którym pracujesz, może się okazać, że niektórzy programiści generują znacznie mniej zmian niż inni pomimo wykonywania podobnej ilości pracy. W takim przypadku z listy *Contributions* (aktywności), znajdującej się obok wykresów aktywności, wybierz opcję *Additions* (dodania)



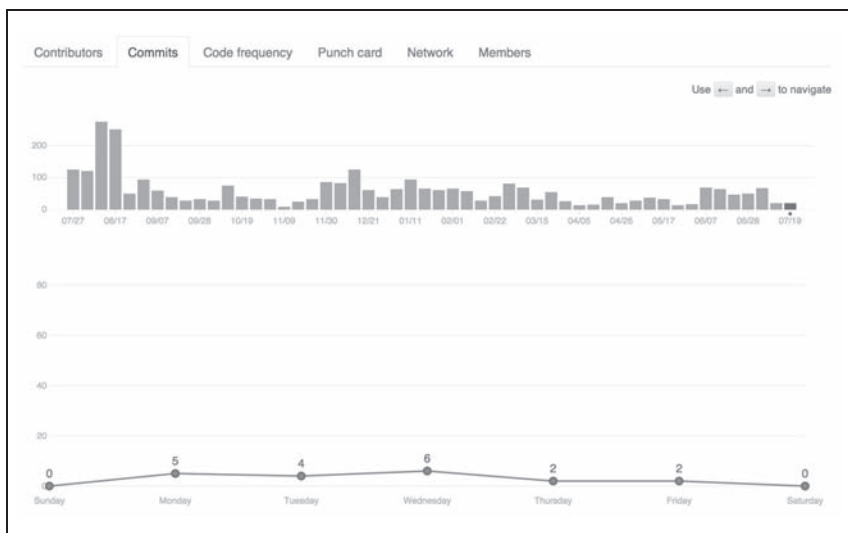
Rysunek 2.11. Wykres aktywności dla lata 2013 roku

albo *Deletions* (usunięcia). W ten sposób będziesz mógł lepiej oszacować liczbę linii kodu dodanych do projektu lub usuniętych z niego przez poszczególnych programistów. W wypadku modyfikacji zmiana w jednej linii zostanie policzona jako usunięcie starej linii i dodanie nowej.

Wykres zmian

Wykres zmian, pokazany na rysunku 2.12, ilustruje liczbę zmian tygodniowo w ciągu całego okresu pracy nad projektem, co pozwala z grubsza zorientować się w aktywności tegoż projektu i jej zmienności w czasie.

Zasadniczym powodem, dla którego warto zapoznać się z wykresem zmian, jest możliwość oszacowania liczby zmian tygodniowo w trakcie całego okresu trwania projektu. Diagram zaczyna się od zestawienia słupkowego, w którym jeden słupek odpowiada jednemu tygodniowi. Takie zestawienie pozwala łatwo wychwycić cykliczność lub długoterminowe trendy. Czy liczba zmian w projekcie stopniowo spada? Czy dokooptowanie nowych programistów spowodowało zauważalny wzrost liczby zmian? Czy większość zmian przypada na ostatni tydzień każdego miesiąca albo czy na wykresie daje się zauważyć tendencje sezonowe? Omawiany wykres pozwala używać dobre rozeznanie w tym, jak liczba zmian — którą można traktować jako bardzo przybliżoną miarę produktywności zespołu — ulega modyfikacji w czasie.



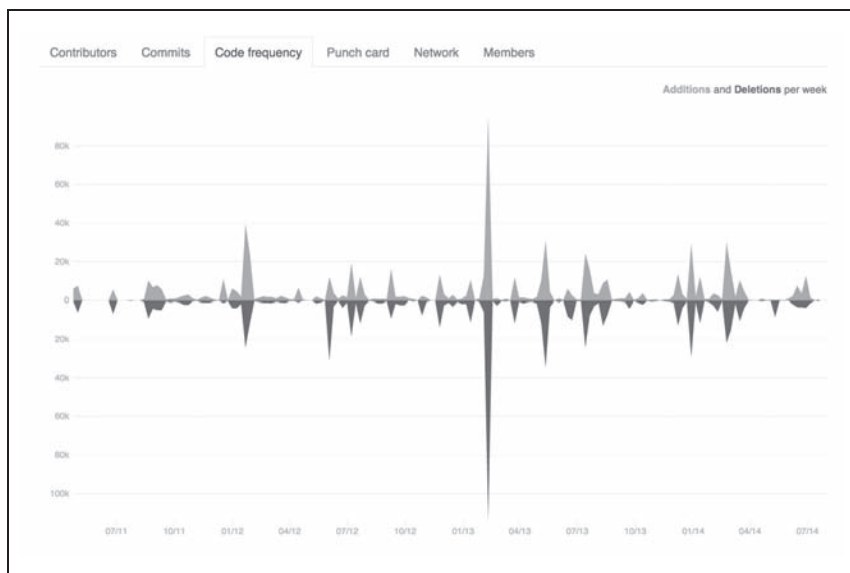
Rysunek 2.12. Wykres liczby zmian w projekcie Bootstrap

Pod zestawieniem słupkowym znajduje się wykres liniowy pokazujący uśrednioną wartość zmian w poszczególnych dniach tygodnia podczas całego życia projektu. Wykres ten może się przydać do oszacowania cyklu aktywności w trakcie typowego tygodnia pracy nad projektem. Czy na przykład uczestnicy nie wysyłają zmian w poniedziałki ze względu na dużą liczbę spotkań? A może większość zmian wprowadzają w czwartki, przed piątkowymi prezentacjami? Albo zbyt dużo pracują w weekendy, co nie rokuje utrzymania tempa rozwoju projektu w dłuższej perspektywie czasowej?

Wykres częstotliwości zmian w kodzie

Wykres częstotliwości zmian w kodzie, zaprezentowany na rysunku 2.13, przedstawia liczbę linii dodanych do projektu i usuniętych z niego w kontekście czasu. Wykres ten doskonale nadaje się do sprawdzania tego, kiedy w ramach pracy nad danym projektem zaszły w nim największe zmiany.

W wypadku refaktoryzacji kodu programiści często dodają i usuwają setki albo nawet tysiące linii naraz, podczas gdy w zwykłym cyklu pracy jedna zmiana obejmuje najczęściej tylko kilka dodanych, usuniętych lub zmodyfikowanych linii. We wspomnianej refaktoryzacji liczba zmian może być relatywnie mała, przy czym liczba zmodyfikowanych linii będzie wyraźnie



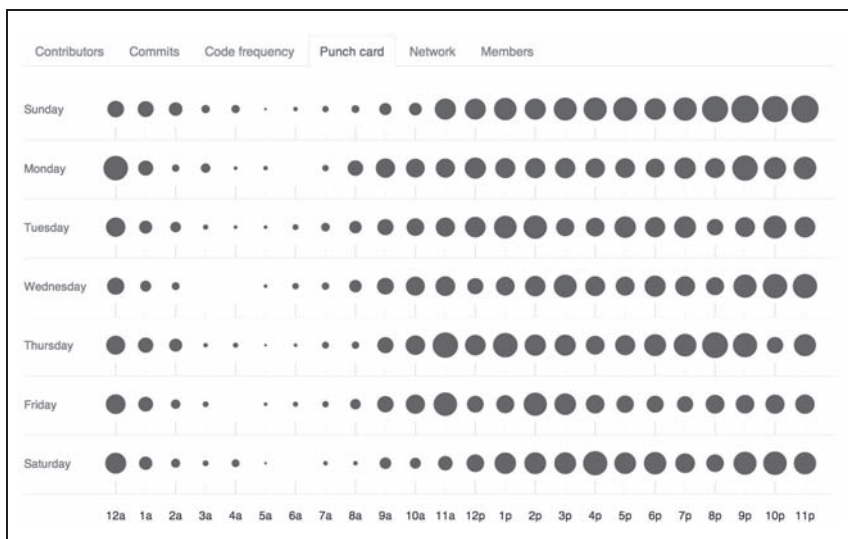
Rysunek 2.13. Wykres częstotliwości zmian w kodzie dla projektu Bootstrap

większa; jeśli więc chciałbyś oszacować, kiedy w projekcie zaszły najważniejsze przeobrażenia, powinieneś zacząć od wykresu częstotliwości zmian w kodzie. Na przykład na podstawie rysunku 2.13 widać, że w lutym i marcu 2013 roku doszło do znaczącej refaktoryzacji kodu.

Wykres „karta perforowana”

Wykres typu „karta perforowana”, przedstawiony na rysunku 2.14, pozwala oszacować, o jakiej porze dnia i w jakich dniach wprowadzono najwięcej zmian.

Na wykresie tego typu poszczególne godziny we wszystkich dniach tygodnia zostały przedstawione w postaci kółek. Średnice kółek są proporcjonalne do procentowej liczby zmian w projekcie, wprowadzonych o danej porze danego dnia. Im większe koło, tym więcej zmian w projekcie nastąpiło właśnie w tym czasie. Jest to kolejny świetny sposób na oszacowanie najbardziej produktywnych pór dnia dla zespołu.



Rysunek 2.14. Wykres „karta perforowana” dla projektu Bootstrap

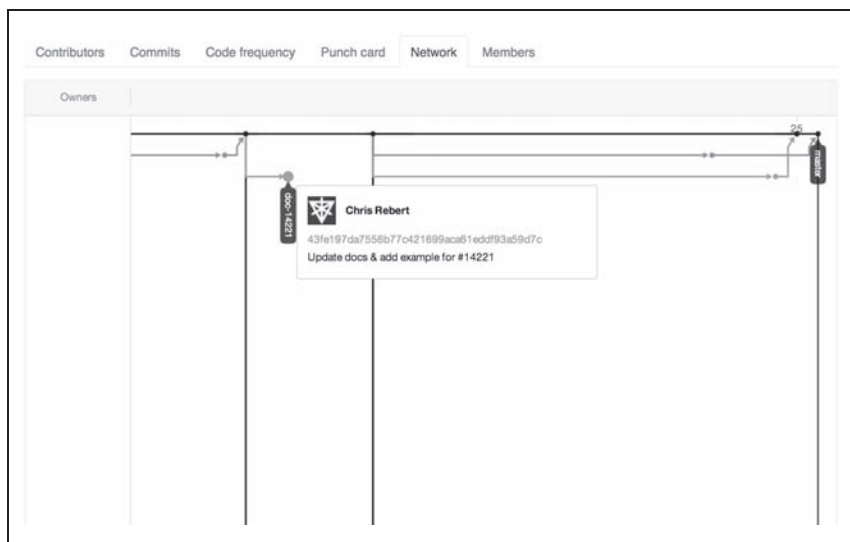
Wykres sieciowy

Wykres sieciowy, zaprezentowany na rysunku 2.15, ilustruje liczbę gałęzi i zmian w tych gałęziach w trakcie życia projektu. Przedstawia on także odgałęzienia całego repozytorium (forki) wykonane przez uczestników.

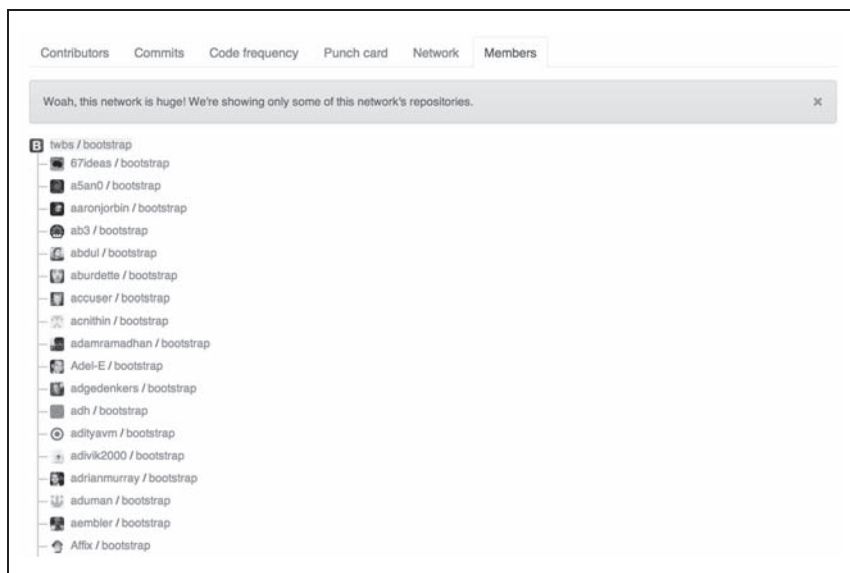
Wykres sieciowy przydaje się do oszacowania postępów prac w danej gałęzi i daje pewien pogląd na rodzaj zadań wykonywanych przez wybranego członka zespołu w obrębie jego forka. Gdy zmiany te trafiają do głównej gałęzi w źródłowym repozytorium, fakt ten jest symbolizowany strzałką z informacją o scaleniu zmiany (jeśli została ona scalona przez zgłoszenie). Wybraną zmianę można wskazać kursorem myszy, co spowoduje wyświetlenie jej autora oraz dołączonego do niej komentarza.

Lista uczestników

Ostatnie zestawienie, które mogą obejrzeć wszyscy, niezależnie od rodzaju uprawnień, to lista uczestników projektu. Jeśli w danym projekcie zrobiono wyjątkowo dużo rozgałęzień (forków), na ekranie pojawi się komunikat taki jak na rysunku 2.16, informujący o wyświetleniu tylko częściowego ich zestawienia.



Rysunek 2.15. Wykres sieciowy

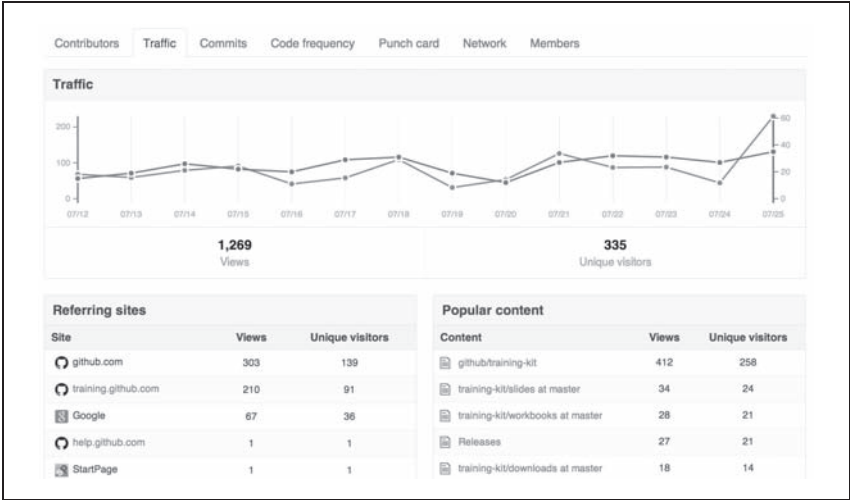


Rysunek 2.16. Lista członków

Lista uczestników zawiera tylko te osoby, które wykonały odgałęzienia repozytorium albo odgałęzienia odgałęzień. Te osoby nie należą do zespołu pracującego nad oryginalnym, źródłowym repozytorium i z tego względu potrzebują jego kopii, by mogły wprowadzać zmiany za pośrednictwem zgłoszeń.

Wykres ruchu sieciowego

Jest jeszcze jeden pomocniczy wykres, dostępny tylko dla właścicieli i współautorów projektu, a mianowicie wykres ruchu sieciowego, pokazany na rysunku 2.17.



Rysunek 2.17. Wykres ruchu sieciowego

Wykres ten przedstawia liczbę wyświetleń projektu i liczbę unikatowych gości strony na przestrzeni czasu. Na stronie z tym wykresem są ponadto podane takie informacje jak adresy stron WWW, z których internauci trafiają na stronę projektu, oraz najpopularniejsze materiały na stronie danego projektu w serwisie GitHub. Prezentowany wykres to świetny sposób na oszacowanie popularności projektów open source.

Na bazie informacji przedstawionych w tym rozdziale powinieneś nieźle się orientować, jak można szybko zapoznać się z projektem na podstawie pliku README, analizy zmian, zgłoszeń, problemów, pulsu i wykresów dostępnych w serwisie GitHub. W następnym rozdziale przyjrzymy się zagadnieniom dotyczącym uczestnictwa w projekcie.

Skorowidz

A

akceptowanie zgłoszenia, 62
API, 88
automatyczne uzupełnianie nazwy, 86
automatyczny generator stron, 87
autoryzacja, 112
autoryzacja dwuetapowa, 116

B

błąd autoryzacji, 112
Bootstrap, 17
branch, gałąź, 14

C

changes, zmiany, 109
check out, weryfikacja, 15
clone, klonowanie, 16
commit message, komentarz
do zmiany, 14
commit, zmiana, 14

D

definiowanie nowego problemu, 65
dodawanie
etykiety, 70
kamienia milowego, 68
klucza dostępu, 94
komentarza, 60
pliku, 34

stron, 74
usługi, 91
użytkownika, 99
współpracowników, 84
zespołu, 98
dokumentacja, 15
dostęp do repozytorium, 94

E

edycja, 33
offline, 102
w IDE, 101
edytor tekstu vi, 123
edytowanie
etykiety, 69
pliku, 44–47, 59
wielu plików, 102
ekran
edytowania pliku, 47
podglądu zgłoszenia, 38
tworzenia zgłoszenia, 39
zgłoszeń, 38
emotikona, 58
etykiety, 15, 68

F

feature branch, gałąź funkcji, 14
filtrowanie usług, 91
folder, 48
główny, 18
tworzenie, 49
zmiana nazwy, 49

- fork, rozgałęzienie projektu, 16
- forking, 11, 33
- formatowanie, 74
- formularz
 - generowania stron, 76
 - nowego problemu, 66
 - repozytorium, 82

G

- gałąź, 14
 - base:master, 54
 - compare:master, 54
 - funkcji, 14
 - gh-pages, 78
 - główna, master, 14, 18, 78
 - new_feature, 110
 - tematyczna, 14
 - update_readme, 53, 54, 59
 - wydania, 14
- generowanie stron WWW, 76
- Git, 11
- GitHub, 11
- główny ekran programu, 118
- GUI, 102

H

- historia zmian, 20, 44

I

- IDE, Integrated Development Environment, 101
- informacje
 - o zespole, 99
 - o zmianie, 41
- informacyjne strony WWW, 75
- inicjalizowanie repozytorium, 83
- inicjowanie zgłoszenia, 54
- instalacja
 - programu, 115
 - w Mac OS, 104

- integracja GitHuba
 - z Asaną, 93
 - z Basecampem, 92
- interfejs
 - graficzny, GUI, 102
 - interfejs programistyczny aplikacji, API, 88
- issue, problem, zadanie, 11, 15, 22, 65

K

- kamienie milowe, 67, 68
- karta perforowana, 29
- klonowanie, 16
 - katalogu, 109
 - repozytorium, 101
- klucz
 - dostępu, 92–94
 - publiczny, 106, 118
- komentarz, 58
 - do zmiany, 14, 60
 - z emotikonami, 58
- komentowanie
 - problemów, 70
 - zgłoszeń, 57
- komunikat o błędzie autoryzacji, 112
- konfigurowanie
 - kluczy dostępu, 93
 - narzędzi, 123
 - repozytorium, 86
 - ustawień Gita, 105, 117
- konto maszynowe, 94
- kopia repozytorium, 34
- kreator
 - instalacji, 104
 - konfiguracji, 116

L

- liczba zgłoszeń, 23
- lista
 - Add service, 91
 - Contributions, 26

- członków, 31
- gałęzi, 52
- Pages, 74
- stron, 75
- uczestników, 30
- zmian, 20

logowanie, 105

M

Mac OS, 103, 107, 112

master branch, gałąź główna, 14, 18, 78

menu Settings, 123

merge, scalanie, 15

merge commit, zatwierdzenie

- zgłoszenia, 42, 44

modyfikowanie zgłoszeń, 59

N

narzędzie

- Bootstrap, 17
- Issues, 70
- Pages, 78

nazwa folderu, 49

nieprzesłane zmiany, 122

nowa zmiana, 61

nowe zgłoszenie, 55

O

obsługa

- emotikonów, 58
- Gita, 123
- problemów i zadań, issues, 11
- zdarzeń, 90, 91

odgałęzienia, 33, 34

odsyłacz

- 4 commits, 44
- Graphs, 25
- Issues, 23
- Pull Requests, 20
- Pulse, 23

Settings, 85, 86

Teams, 97

ograniczenia edytowania, 49

opcja Markdown, 73

opcje programu GitHub, 124

options, opcje, 123

organizacja, 95

P

plik

- .gitkeep, 49
- new_file.md, 38
- README.md, 19, 44–46, 53, 83
- windows_feature.html, 121

pobieranie, 101

podgląd, 73

- gałęzi, 113
- repozytorium, 121
- strony, 74
- ustawień, 113
- zmian, 111

pole

- Edit Message, 74
- Filter services, 91
- Ignored files, 114
- Summary, 111, 122

polecenie Options, 123

potwierdzenie usunięcia

- repozytorium, 89

powiadomienia o zgłoszeniach, 63

praca

- z folderami, 48
- z wiki, 72

preview, podgląd, 73

problem, 15, 22–24, 65

program GitHub.app, 108

przeglądanie

- historii zmian, 20
- projektu, 17
- zgłoszeń, 22, 57

przenoszenie pliku, 47

- przesyłanie
 - repozytorium, 85
 - zmian, 110, 121
- przycisk
 - Add collaborator, 85
 - Add service, 91
 - Add webhook, 90
 - Automatic page generator, 87
 - branch:master, 51
 - cancel, 48
 - Clone, 109
 - Clone in Desktop, 107, 108, 118, 119
 - Comment, 57
 - Commit, 112
 - Create pull request, 55
 - Create repository, 83
 - Create team, 98
 - Delete this repository, 88
 - Done, 106
 - Download GitHub for Windows, 114
 - Edit, 45, 69, 74
 - Fork, 33
 - Install, 114
 - Install Command Line Tools, 106
 - Labels, 68
 - Launch Application, 109
 - Log in, 116
 - Merge pull request, 41, 62
 - Milestones, 67
 - New issue, 65, 66
 - New milestone, 67
 - New Page, 74
 - New pull request, 37
 - Remove, 99
 - Rename, 86
 - Settings, 76, 123
 - Submit new issue, 67
- pull requests, zgłoszenia, 11, 15, 20, 33, 56
- puls, 23, 24

R

- release branch, gałąź wydania, 14
- repo, 34
- repozytoria
 - organizacji, 95
 - osobiste, 95
- repozytorium, 81
 - Bootstrapa, 18
 - PeterBell, 40
 - single-repo-example, 52
- root, 18
- rozgałęzienie projektu, 16
- rozwidlenie, 16
- ruch sieciowy, 32

S

- scalanie, 15
- sekcja
 - Add a deploy key, 94
 - Danger Zone, 88
 - GitHub Pages, 78
 - Settings, 87
 - Uncommitted changes, 122
- settings, ustawienia, 76
- strona
 - Add webhook, 90
 - główna organizacji, 97
 - Labels, 69
 - projektu, 17
 - Settings/Options, 87
 - Teams, 97, 99
 - wiki, 73
- summary, streszczenie, 122
- system
 - kontroli wersji, 11
 - Mac OS, 102
- szablon, 77

Ś

- środowiska programistyczne, 101

T

- tag, etykieta, 15
- testowanie zgłoszenia, 61
- tworzenie
 - dokumentacji, wiki, 11
 - folderu, 49
 - gałęzi, 122
 - i konfigurowanie, 81
 - klucza dostępu, 94
 - kopii repozytorium, 34
 - kopii zapasowej, 101
 - nowego pliku, 35, 36
 - organizacji, 95, 96
 - repozytorium, 79, 81
 - rozgałęzień projektu, forking, 11
 - strony internetowej, 76, 78
 - strony w wiki, 73
 - zespołu, 98
 - zgłoszenia, 36, 39, 54

U

- uatrakcyjnianie komentarzy, 58
- udział w projekcie, 33
- umieszczanie odsyłacza, 74
- uruchamianie programu, 102, 109, 119
- usługi, 90, 91
- ustawienia zespołu, 100
- usuwanie repozytorium, 88

W

- weryfikacja, 15
- wiersz poleceń, 106, 123
- wiki, 11, 15, 71
- Windows, 114
- właściciel repozytorium, 82
- współpraca, 51
- wykonywanie testów, 102
- wykres
 - aktywności, 27
 - częstotliwości zmian, 28, 29

- liczby zmian, 28
- ruchu sieciowego, 30–32
- współpracowników, 25, 26
- zmian, 27

WYSIWYG, 72

- wysyłanie zgłoszeń, pull requests, 11
- wyświetlanie
 - pliku README.md, 19, 45
 - problemów, 22
 - pulsu, 23
 - wykresów, 25
 - zespołów, 98
 - zgłoszenia, 42, 43
 - zmian, 112

Z

- zadanie, 15
- zakładka
 - Changes, 109
 - Collaborators, 86
 - Deploy keys, 92
 - Issues, 67
 - Options, 86–88
 - Preview, 73
 - Pull Requests, 37, 54, 57
 - Settings, 113
 - z ustawieniami, 79
- zalecenia dotyczące
 - problemów, 70
 - zgłoszeń, 64
- zalety
 - Gita, 12
 - GitHuba, 13
- zarządzanie
 - etykietami, 68
 - kamieniami milowymi, 67
 - zespołami, 96
- zatwierdzenie zgłoszenia, 42, 44
- zdalne konto, 106
- zespół, 96
- zespół Owners, 97

zgłoszenia 15, 20–22, 33, 36–40, 54, 59
 akceptowanie, 62
 komentowanie, 57
 modyfikowanie, 59
 opinie, 56
 powiadomienia, 63
 przeglądanie, 57
 testowanie, 61
 tworzenie, 54
 z gałęzi, 60
 zalecenia, 64
 zmian, 46

zmiana, 14
 do przesłania, 122
 nazwy folderu, 49
 nazwy pliku, 47
 początkowa, 53
 w odgałęzieniu, 51
zmienianie kodu, 28
znaczniki, 73, 75
znak
 @, 56
 krzyżyka, 70
 ukośnika, 47
znaki ../, 48

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

GRUPA WYDAWNICZA



Helion SA

GitHub. Przyjazny przewodnik

Git zrewolucjonizował świat systemów kontroli wersji. Koncepcja rozproszonego systemu podbiła serca programistów i ustanowiła trendy na kolejna lata. Co powstało z połączenia Gita i serwisu udostępniającego miejsce oraz narzędzia dla programistów korzystających z Gita? To GitHub – najpopularniejsze miejsce w sieci, w którym znajdziesz tysiące projektów. GitHub wytycza dziś ścieżki dla platform ułatwiających pracę z rozproszonymi systemami kontroli wersji.

Ta książka została w całości poświęcona właśnie GitHubowi. Sięgnij po nią i przekonaj się, jakie usługi oferuje ten serwis oraz jak wykorzystać jego potencjał w Twoim projekcie. W trakcie lektury dowiesz się, jak przeglądać udostępniony kod, współpracować przy projektach oraz dostarczać poprawki. Ponadto nauczysz się zarządzać zgłoszonymi przez użytkowników problemami w Twoim kodzie oraz opanujesz najlepsze praktyki konfigurowania Twojego repozytorium. Na sam koniec poznasz narzędzia przeznaczone dla systemów Mac OS i Windows, a służące do współpracy z serwisem GitHub. Książka ta jest obowiązkową lekturą dla wszystkich czytelników chcących w pełni wykorzystać potencjał drzemiący w tym serwisie.

Dzięki tej książce:

- poznasz system kontroli wersji Git
- stworzysz własne repozytorium w serwisie GitHub
- poznasz najlepsze techniki pracy z GitHubem
- wykorzystasz w pełni możliwości GitHuba

Zaawansowany system kontroli wersji dla każdego!

Peter Bell – przedsiębiorca, Agile Coach, założyciel Ronin Labs i dyrektor ds. technicznych tej organizacji, trener Gita. Regularnie występuje na konferencjach, prezentuje tematy związane z Ruby, Node.js, chmurami obliczeniowymi i językami programowania. Jego artykuły były publikowane w takich magazynach, jak „Dr. Dobbs”, „IEEE Software” i „GroovyMag”.

Brent Beer – pasjonat Gita. Wierny fan języka Ruby. Od 2011 roku związany z GitHubem, obecnie jest trenerem w tym serwisie.



33595 numer katalogowy
księgarnia internetowa

<http://helion.pl>

zamówienia telefoniczne



0 801 339900



0 601 339900

Sprawdź najnowsze promocje:

🔗 <http://helion.pl/promocje>

Książki najchętniej czytane:

🔗 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

🔗 <http://helion.pl/nowosci>

Helion SA

ul. Kościuszki 1c, 44-100 Gilwice

tel.: 32 230 98 63

e-mail: helion@helion.pl

<http://helion.pl>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-283-0503-8



9 788328 305038

Informatyka w najlepszym wydaniu

cena 32,90 zł