

Uprawnienia / Analiza rozwiązań

Streszczenie	3
Cel	3
Sposób realizacji	3
Konkluzja	3
Kontekst badawczy	4
Analogiczne rozwiązania	4
Najwygodniejszy model uprawnień	4
Przegląd rozwiązań	5
Źródła	18

Streszczenie

Cel

Przegląd systemów uprawnień w kilku popularnych środowiskach programistycznych. Intencją przeglądu było zebranie rozwiązań, które znalazły już bardzo szerokie zastosowanie i — jako takie — mogą być uznane za wygodne i od dawna zaakceptowane przez użytkowników. Chcemy wzorować się na nich.

Założenia wstępne dla systemu uprawnień w aplikacji.gov.pl:

- Wprowadzamy uprawnienia tylko na poziomie całych aplikacji na instancji platformy
- Nie proponujemy wspólnego modelu ani mechanizmu uprawnień wewnątrz aplikacji
- Aplikacje — każda wewnętrznie zarządza prawami do jej operacji.
- "Właściciel" instancji platformy ma prawo udzielić praw do zarządzania całym aplikacjami innym osobom.

Sposób realizacji

Wykonano przegląd dokumentacji popularnych platform / serwisów społecznościowych:

Nie braliśmy pod uwagę rozbudowanych rozwiązań opartych na LDAP, takich jak Microsoft Active Directory, gdyż wymagają one dedykowanej obsługi operatorskiej a ich skomplikowanie przekracza zakres planowanych uprawnień w aplikacji.gov.pl.

Konkluzja

Jeżeli przyjmiemy, że zgodnie z założeniem wstępnym, na platformie nie udostępniamy wspólnego mechanizmu przechowywania i kontroli uprawnień specyficznych dla aplikacji, to jedynym mechanizmem, który będzie wspólny dla wielu aplikacji jest SSO — wspólne, jednokrotne logowanie do aplikacji. Chcemy jednak jak najbardziej ułatwić korzystanie z platformy w kontekście wykonywanej operacji.

Rozumiemy to w ten sposób, że użytkownik który pracuje w jakiejś aplikacji (gdzie ma odpowiednie uprawnienia), jeśli — np. za pomocą myszy — przeniesie swoje zainteresowanie (tzw. focus) na widoczne w innym miejscu ekranu przyciski związane z inną aplikacją, to płynnie przejdzie do operacji w tej aplikacji, a mechanizm SSO spowoduje, że nie będzie musiał się dodatkowo logować. Podobnie, próba wykonania operacji na poziomie samej platformy, będzie możliwa w sposób płynny (oczywiście. o ile użytkownik ma odpowiednie uprawnienia).

Model uprawnień, który zapewnia taką pracę, powinien być zbliżony do (opisanego w dalszej części) modelu amazon.com.

Kontekst badawczy

Analogiczne rozwiązania

Badano rozwiązania schematów praw użytkowników istniejące w:

Amazon

Facebook

Atlassian Confluence

Atlassian JIRA

Poniżej najdokładniej opisano mechanizm uprawnień istniejący w Atlassian JIRA, gdyż jest to system, który wartoby naśladować, gdyby był realizowany wariant aplikacji.gov.pl, w którym platforma przechowywałaby wspólny dla wielu aplikacji schemat uprawnień, oparty o wspólny schemat "ról".

Ponieważ zapewne będzie realizowany wariant bez takiego wspólnego repozytorium szczegółowych uprawnień, wydaje się, że najbliższy potrzeb aplikacji.gov.pl jest model stosowany przez amazon.com.

Najwygodniejszy model uprawnień

W tabelce poniżej zestawiono cechy badanych platform i zaznaczono model najbardziej zbliżony do potrzeb aplikacji.gov.pl:

- niebieską elipsą - gdyby planowano przechowywanie na platformie złożonych uprawnień poszczególnych aplikacji
- zieloną elipsą - jeśli każda aplikacja osobno pamięta swoje schematy uprawnień

	Microsoft Active directory	Amazon	Facebook	Atlassian Cofluence	Atlassian JIRA
Szeroko znany	+	+	+	+	+
Prosty	-	+	+	+	+ / -
Wygodny dla organizacji z pracownikami	+	+	-	-	+
Umożliwia przechowanie złożonych uprawnień	+	-	-	-	+

Przegląd rozwiązań

Amazon

Użytkownicy z każdej firmy wykorzystującej platformę Amazon, mogą mieć przypisane role ze zbioru: Administrator, Marketer, Developer, Analyst. Każda z ról ma własny zestaw uprawnień. Wybierając dla każdego użytkownika odpowiednie role, właściciel konta może sterować uprawnieniami swoich pracowników.

Role:

- Administrator — zarządzanie uprawnieniami użytkowników i informacjami o płatnościach; edycja informacji o profilu firmy
- Marketer — edycja informacji o detalach aplikacji; dostęp do raportów sprzedaży
- Developer — ładowanie APKs i plików binarnych z aplikacjami firmy;
- Analyst — dostęp do raportów o płatnościach i zyskach

Elementarnymi uprawnieniami są:

- App files information - pobieranie i ładowanie plików aplikacji
- App details - edycja informacji o szczegółach aplikacji
- Company profile - modyfikowanie informacji o profilu firmy
- Earnings/Payments Reports - dostęp do raportów o płatnościach i zyskach
- Help - pomoc
- My account - informacja o własnym koncie
- Sales reports - raporty sprzedaży
- User permissions - zmiany uprawnień użytkowników

User Permissions

[Hide permissions matrix](#)

	Administrator	Marketer	Developer	Analyst
App files information	✓		✓	
App details	✓	✓		
Company profile	✓			
Earnings/Payments reports	✓			✓
Help	✓	✓	✓	✓
My account	✓	✓	✓	✓
Sales reports	✓	✓		✓
User permissions	✓			

Facebook

Każde zezwolenie ("Permission") ma własny zestaw wymagań ("requirements") i ma zalecane ("suggested") przypadki użycia. Niektóre zezwolenia nie wymagają przeglądu ("review"), ale większość tak. Użycie tych zezwoleń jest przedmiotem "Platform Policies" i prywatnych reguł dostępu. (W dokumentacji są podane także "Common Usage")

Uprawnienia:

- `public_profile` - (tylko ten nie wymaga Client OAuth Login - zezwolone w App Dashboard --> Facebook Login)
- `user_friends` - dostęp do listy przyjaciół ("friends") używających tej samej aplikacji; lista w `user->friends_edge`
- `email` - dostęp do głównego ("primary") e-adresu, poprzez obiekt `user.email`
- `user_about_me` - opis osobisty (to co w Profile --> About Me); via obiekt `user`
- Uwaga: nie daje dostępu do danych publicznych (dostępnych via `user.public_profile`);
- Facebook kontroluje sposób użycia
- `user_actions.books` - akcje związane z wszystkimi książkami opublikowanymi przez dowolną aplikację; obejmuje książki przeczytane, poszukiwane, oceniane lub cytowane; Facebook kontroluje
- `user_actions.fitness` - akcje związane z wszystkimi akcjami Open Graph fitness opublikowanymi przez dowolną aplikację; obejmuje biegi, spacer i jazdę na rowerze; Facebook kontroluje
- `user_actions.music` - akcje związane z wszystkimi akcjami muzycznymi Open Graph opublikowanymi przez dowolną aplikację; obejmuje słuchane piosenki i tworzone listy; Facebook kontroluje
- `user_actions.news` - akcje związane z wszystkimi akcjami nowinek ("news") Open Graph opublikowanymi przez dowolną aplikację; obejmuje artykuły czytane i publikowane; Facebook kontroluje
- `user_actions.video` - akcje związane z wszystkimi akcjami opublikowanymi przez dowolną aplikację; Facebook kontroluje
- `user_actions:{app_namespace}`

- user_birthday
- user_education_history
- user_events
- user_games_activity
- user_hometown
- user_likes
- user_location
- user_managed_groups
- user_photos
- user_posts
- user_relationships
- user_relationship_details
- user_religion_politics
- user_tagged_places
- user_videos
- user_website
- user_work_history
- read_custom_friendlists
- read_insights
- read_audience_network_insights
- read_page_mailboxes
- manage_pages
- publish_pages
- publish_actions
- rsvp_event
- pages_show_list
- pages_manage_cta
- pages_manage_instant_articles
- ads_read
- ads_management
- business_management
- pages_messaging
- pages_messaging_subscriptions
- pages_messaging_payments
- pages_messaging_phone_number

Uprawnienia przestarzałe

- manage_notifications
- read_stream
- read_mailbox
- user_groups
- user_status

Atlassian — Confluence

(Uprawnienia są odmiennie zorganizowane w Confluence i w JIRA.)

Prawa może mieć

- użytkownik
- grupa użytkowników
- użytkownik anonimowy.

Przestrzeń ("Space")

- jest fragmentem zasobów zarządzanych przez Confluence
- ma osobne prawa.

Prawo "Space Admin" do danej "Space", automatycznie uzyskuje jej twórca. Może następnie to prawo przypisać lub odbierać innym użytkownikom lub grupom.

Sprawdzanie cech "Space", m.in. twórca, administratorzy, strona główna WWW itp.:
Confluence header --> Spaces --> Space directory --> Space Details icon

Zmiana uprawnień do "Space"

Space tools --> Permissions from the bottom of the sidebar --> Edit Permissions

Uprawnienia:

- View — przeglądanie całej zawartości "Space"
- Add — tworzenie i edycja stron, blogów, komentarzy i załączników
- Restrict — możliwość ograniczenia innym użytkownikom prawa oglądania lub edycji
- Delete own — możliwość usunięcia strony, blogu lub załącznika utworzonego przez siebie, niezależnie czy inni użytkownicy edytowali ten obiekt
- Delete — możliwość usuwania stron, komentarzy, załączników i listów
- Export — możliwość eksportu całej przestrzeni ("Space"). (Uwaga: prawo do eksportu pojedynczego dokumentu ma każdy kto ma prawo przeglądać)
- Admin — prawo administracji całą "Space", obejmujące przekazanie tego prawa innym

Administratorzy "Confluence" niekoniecznie są administratorami "Space". Aby się stać "Space Admin" jakiejś "Space", należy otrzymać uczestnictwo w grupie która ma prawa "Admin" i "View" dla tej "Space" albo otrzymać wprost te dwa uprawnienia.

Osoby posiadające globalne uprawnienie "System Administrator" mogą sobie nadać prawa do dowolnej "Space".

Uprawnienia do "Space" są addytywne — każdy ma uprawnienia które ma on nadane osobiście lub które posiada którakolwiek z grup, do której należy.

Atlassian — JIRA

Zarządzanie uprawnieniami

Uprawnienia projektowe są tworzone wewnątrz pewnych „schematów uprawnień”, które następnie są przypisywane do projektów.

Uprawnienia projektowe mogą być udzielane

- Pojedynczym osobom
- Grupom użytkowników
- „Rolom” projektowym
- „Ogólnym rolom” (ang. "Issue roles") takim jak „Sprawozdawca”, „Lider projektu”, „Bieżący przypisany”
- Komukolwiek (aby dopuścić osoby niezalogowane)

Poza tym możliwe jest jednoczesne udzielenie uprawnień wielu użytkownikom lub wielu grupom, przez wskazanie ich cech.

Niektóre uprawnienia zależą od innych, upewniając się, że użytkownik ma prawa do potrzebnych akcji. Np. aby użytkownik mógł zatwierdzić "issue", musi mieć zarówno uprawnienie "Transition Issue", jak i uprawnienia "Resolve Issue".

Przegląd uprawnień projektowych

- Project permissions
 - Administer projects
 - Browse projects
 - Manage sprints (only available to JIRA Software users)
 - View development tools (only available to JIRA Software users)
 - View (read-only) workflow
- Issue permissions
 - Assign issues
 - Assignable user
 - Close issues
 - Create issues
 - Delete issues
 - Edit issues
 - Link issues
 - Modify reporter
 - Move issues
 - Resolve issues
 - Schedule issues
 - Set issues security
- Voters & watchers permissions
 - Manage watcher list
 - View voters and watchers
- Comments permissions
 - Add comments
 - Delete all comments
 - Delete own comments
 - Edit all comments
 - Edit own comments
- Attachments permissions
 - Create attachments
 - Delete all attachments
 - Delete own attachments
- Time-tracking Permissions
 - Work on issues
 - Delete all worklogs
 - Delete own worklogs
 - Edit all worklogs
 - Edit own worklogs

Schemat uprawnień

Jest to kolekcja przypisań użytkowników/grup/ról do pewnych uprawnień projektowych ("project permissions") wymienionych powyżej. Każdy projekt ma schemat uprawnień. Ten sam schemat może być przypisany do wielu projektów.

Schematy uprawnień są przydatne dlatego, że dla wielu organizacji, w wielu projektach — w odniesieniu do uprawnień, są formułowane te same wymagania. Schematy pozwalają unikać wielokrotnego definiowania tej samej struktury uprawnień.

Ujmując te schematy schematycznie (;-)), mamy

```
<Lista Schematów> ::= [ <Schemat uprawnień> ] +  
<Schemat uprawnień> ::= [ <Uprawnienie> | <Użytkownik> | + ] +  
                           | <Grupa>          |  
                           | <Rola>          |
```

Powyższa relacja ma charakter wiele-do-wielu i może być określana w różny sposób.

W narzędziach Atlassian wykonywane jest to w kolejności:

wybierz uprawnienie --> wybierz użytkowników / grupy /role którym to prawo ma być przyznane.

Zaleca się jak najszersze stosowanie przypisań uprawnień do ról (a nie do konkretnych użytkowników lub ich grup). Przypisując następnie konkretnych użytkowników do ról projektowych, uzyskuje się najbardziej przejrzysty i elastyczny schemat uprawnień, w którym jest niewiele przypisań.

Mając gotowy schemat uprawnień, można zmodyfikować opis projektu, przypisując doń wybrany schemat. Atlassian ułatwia zarządzanie schematami uprawnień, udostępniając łatwe operacje kopiowania i usuwania.

Zarządzanie rolami

To elastyczna metoda przypisywania użytkowników albo ich grup do wykonywania pewnych czynności w projekcie.

Role projektowe są definiowane przez administratorów JIRA i wszystkie są dostępne do wszystkich projektów. Administratorzy projektów (ci mający uprawnienie 'Administer Project'; nie muszą mieć globalnego uprawnienia 'JIRA Administrator') — przypisują role w projektach wskazanym użytkownikom.

Role projektowe mogą być używane w:

- permission schemes
- email notification schemes
- issue security levels
- comment visibility
- workflow conditions

Role projektowe mogą mieć również przypisane prawa dostępu do:

- issue filters
- dashboards

Role projektowe są poniekąd podobne do grup użytkowników. Główną różnicą jest, iż cecha należenia do grupy jest globalna, a rola jest specyficzna dla projektu. Konsekwentnie, do modyfikacji grup są uprawnieni administratorzy JIRA, a przypisywanie ról jest uprawnieniem administratorów projektów.

Każdy projekt ma rolę lidera projektu ("project lead"), a każdy komponent ma lidera komponentu ("component lead"). Te role mogą być używane w schematach, sprawach i przepływach, tak samo jak role projektowe.

Po utworzeniu roli, można w wybranym schemacie uprawnień, dokonać asocjacji tej roli z pewnymi uprawnieniami. Również, należy nadać pewnym osobom tę rolę. Od tego momentu, we wszystkich projektach do których przypisany jest wskazany schemat uprawnień, osoby mające tę rolę będą uprawnione do wskazanych czynności.

Uwaga: ponieważ role są globalne, usunięcie jakiejś roli spowoduje zabranie tej roli wszystkim członkom zespołów we wszystkich projektach. Konsekwentnie, spowoduje też zabranie im wszystkich uprawnień z tej roli wynikających — chyba że wynikają one też z innych ról lub zostały osobiście przypisane tym osobom.

Zarządzanie rolami w projekcie

Administrator projektu ("project administrator") ma prawo przypisywać i kasować role osobom w projekcie.

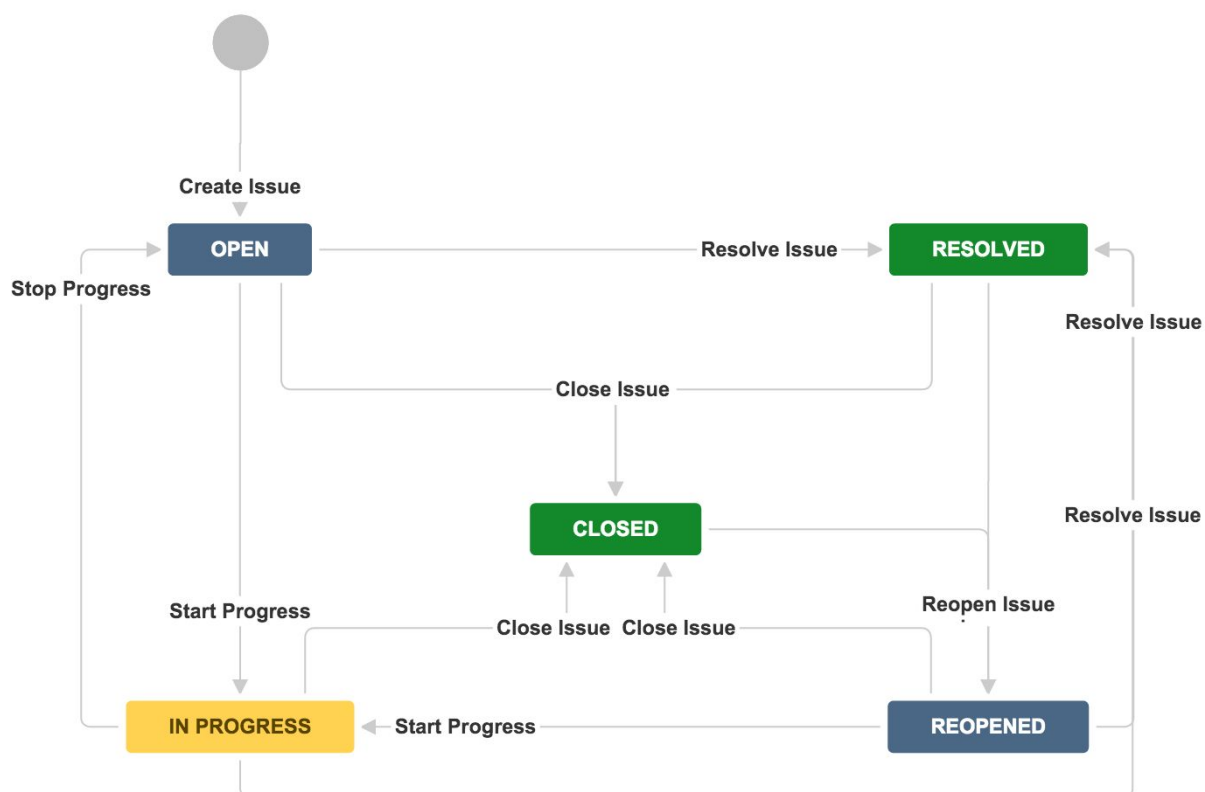
Rola istniejąca w projekcie może nie być przypisana do nikogo; może się wtedy zdarzyć (jeśli schemat uprawnień tego projektu posługuje się rolami), że do jakiejś czynności w tym projekcie nie będzie uprawniony żaden członek zespołu.

Przepływy

Przebieg ("JIRA workflow") to zbiór „statusów” i „przebiegów” ("transitions") przez które sprawa ("issue") przechodzi w swoim cyklu życia. Niektóre przepływy są wbudowane w oprogramowanie Atlassian i nie mogą być zmieniane, ale można je kopiować, by uzyskać własne.

Można też tworzyć własne przepływy od początku albo importować je z Atlassian Marketplace. Przepływy można kojarzyć ze wskazanymi projektami i ewentualnie z rodzajami spraw ("issue types"), używając schematów przepływów ("workflow scheme").

Oto przykładowy przebieg



"Status" reprezentuje specyficzny stan sprawy w pewnym punkcie przepływu. W jakimś momencie, sprawa może mieć tylko jeden status. Opcjonalnie, status może być powiązany z "właściwościami" ("properties"). "Przejście" jest połączeniem między dwoma statusami, które umożliwi sprawie zmianę statusu. Przejścia są jednokierunkowe.

Nieaktywne przepływy (nie używane w żadnym projekcie) mogą być dowolnie edytowane. Edycja przepływu aktywnego tworzy nowy przepływ, który po zatwierdzeniu, może zastąpić oryginalny.

W JIRA istnieje "Workflow designer" — graficzne narzędzie do przeglądania / tworzenia / edycji przepływów.

Praca w trybie tekstowym

Procedury podstawowe

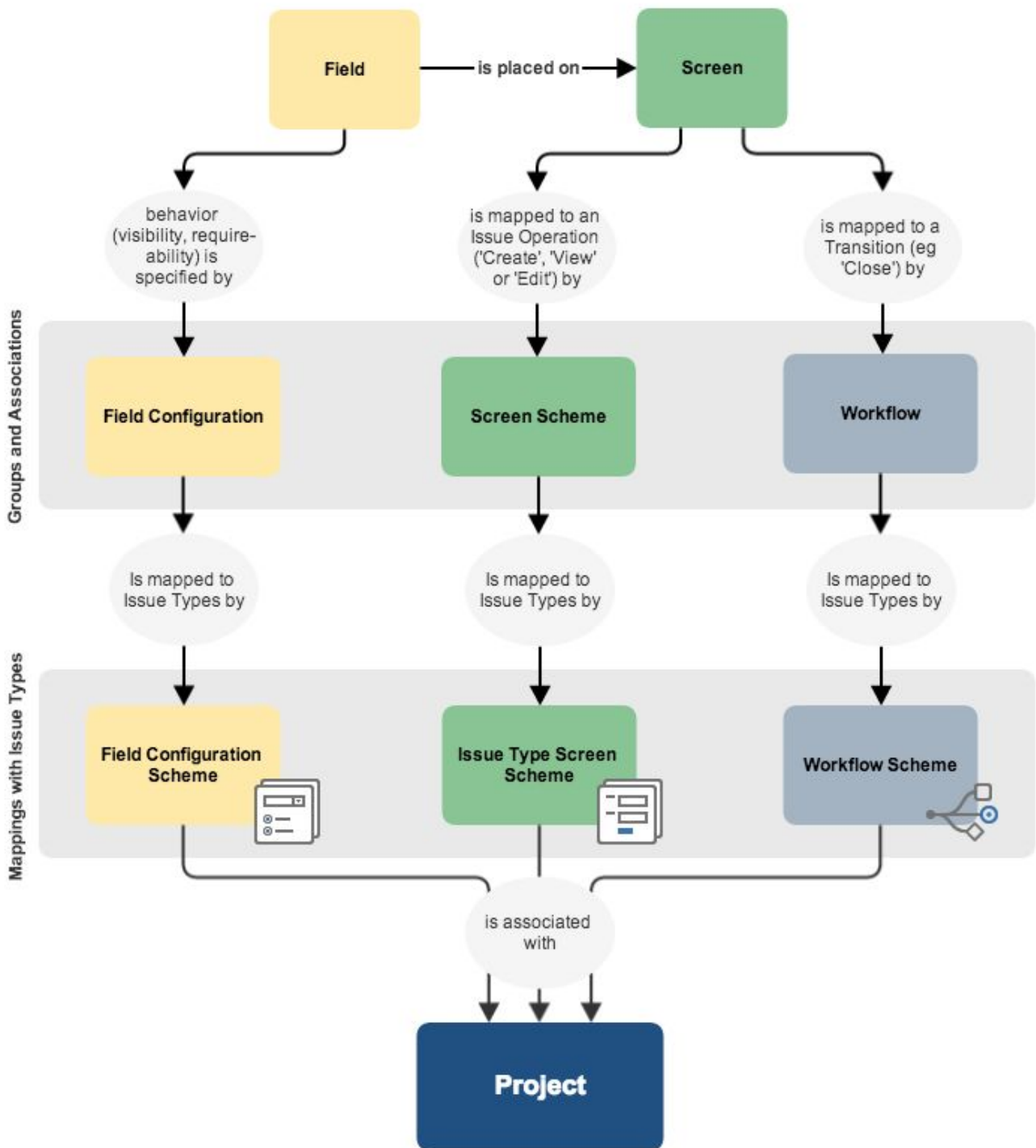
Edycja "kroku"

- Add Transition — dodanie przejścia wychodzącego do kroku
- Delete Transitions — usunięcie jednego lub wielu przejść wychodzących
- Edit — edycja nazwy kroku lub jego statusu
- View Properties — przeglądanie i edycja cech kroku
- Delete Step — usunięcie kroku; możliwe tylko wtedy, gdy krok nie ma przychodzących przejść

Konfigurowanie spraw ("issue")

Aby przystosować JIRA do potrzeb organizacji, można zmieniać wygląd i zachowanie pól spraw ("field issues"), takich jak 'Summary', 'Description', 'Issue Type', etc. Można

- zmieniać opis pola
- czynić je widocznym lub nie
- czynić je wymaganym lub opcjonalnym.
- dodawać własne wartości
 - typu sprawy
 - priorytetu
 - rozdzielczości ("resolution")
 - statusu
- aktywować obsługę otagowanego tekstu ("rich text renderer")
- pozycjonować pola na ekranie
- wybierać, który ekran powinien być wyświetlany dla każdej operacji sprawy (np. 'Create Issue', 'Edit Issue') lub przejścia przepływu ("workflow transition"), np. 'Resolve Issue', 'Close Issue'.



Konfigurowanie schematów przepływów

Schemat przepływu to kolekcja asocjacji (odzworowań) między przepływem i typem sprawy ("issue"). Schematy przepływów są przypisywane do projektów i umożliwiają użycie różnych przepływów dla różnych kombinacji projektu i typu sprawy.

Domyślnie, projekty używają systemowego przepływu JIRA. Ten domyślny schemat przepływu

- przypisuje systemowy przepływ do wszystkich typów spraw, dostępnych w projekcie

- pojawia się jako domyślny schemat przepływu dla wybranego typu projektu.

Istniejący schemat przepływu może być współdzielony przez kilka projektów (zmiana w jednym dotknie wszystkich).

Określanie wartości pól typów spraw ("issue type field values")

JIRA ma zainstalowane domyślny zbiór typów spraw, który może być edytowany, by dopasować do potrzeb zespołu. Można

- kontrolować zbiór typów spraw dostępnych dla każdego projektu
- sterować kolejnością, w jakiej użytkownikowi pojawiają się dostępne typy spraw i jaki jest domyślny typ sprawy dla projektu
- przypisać do poszczególnych typów spraw - specyficzne typy pól, ekranów i przepływów

Uwaga: najprostszą metodą konfiguracji przepływów / projektowania ekranów dla istniejącego typu sprawy w projekcie jest użycie strony "administracji projektu".

Przypisanie do typu sprawy - zachowań pól

Schemat konfiguracji pól ("field configuration scheme") przypisuje konfiguracje pól ("field configurations") do typów spraw w projekcie. Ten sam schemat konfiguracji pól może być użyty w wielu projektach. W projekcie, różne sprawy mogą mieć różne schematy konfiguracji pól.

Zarządzanie przepływami

Przepływy muszą być aktywowane by mogły być użyte, tzn. należy kolejno wykonać

- przypisanie przepływu do schematu przepływów, a potem
- przypisanie schematu przepływów do projektu.

Przypisywanie typów spraw do projektów

Schemat typu sprawy ("issue type scheme") - określa podzbiór typów spraw, który

- ogranicza zbiór typów spraw dostępny dla projektu
- steruje kolejnością dostępnych typów spraw na listach pokazywanych użytkownikom.

Zdefiniowany schemat typu sprawy może być użyty w wielu projektach. Na przykład w organizacji mogą istnieć

- projekty rozwoju oprogramowania
- projekty utrzymaniowe.

Można więc utworzyć "Typ schematu rozwoju oprogramowania", ze sprawami takimi jak "Błąd", "Nowa funkcjonalność" i inny "Typ schematu utrzymaniowego", ze sprawami takimi jak "Zapytanie", "Żądanie zmian".

Te schematy będą mogły być przypisane do wielu konkretnych projektów.

Tworzenie / administracja projektu

Projekt JIRA - to kolekcja spraw ("issue").

Każdy projekt ma specyficzny zbiór cech ("features"). Wszyscy użytkownicy instancji JIRA widzą wszystkie projekty, ale jakie ich cechy i jakie operacje mają dostępne - zależy od uprawnień specyficznych dla projektu.

Z projektem są związane

- schemat uprawnień
- schemat notyfikacji
- schemat bezpieczeństwa spraw
- schemat przepływu
- schemat typów spraw
- schemat ekranów dla typów spraw
- schemat konfiguracji pól.

Szczegóły projektu:

- "klucz projektu" będzie używany jako prefiks kluczy spraw tego projektu (np. 'TEST-100')
- "lider projektu" jest unikalną rolą w projekcie.

Typ projektu można w razie potrzeby zmienić.

Globalne uprawnienia JIRA

Uprawnienia:

- JIRA System Administrators - wszystko
- JIRA Administrators - większość
ograniczenia mogą dotyczyć dostępu do aplikacji
- Browse Users - przeglądanie listy użytkowników i @wywoływania użytkowników w sprawach.
- Create Shared Objects - prawo współdzielenia filtru lub "dashboard" globalnie lub dla wybranych grup użytkowników.
Prawo tworzenia "agile board"
- Manage Group Filter Subscriptions - Prawo zarządzania (tworzenia/usuwania)
"group filter subscriptions"
- Bulk Change - prawo zmian masowych
masowa edycja *
masowe przesunięcie przejść ("bulk workflow Transition")
masowe usunięcie *
(Te zmiany są jeszcze warunkowane uprawnieniami do projektów)

Zarządzanie komponentami

Komponenty to działy projektu. Używane są do grupowania spraw w projekcie. Można do nich przypisywać osoby, o uprawnieniach do wykonywania operacji.

Uprawnienia do komponentów

- Add — tworzenie nowego komponentu
- Select a default assignee — wybór, do kogo sprawy ("issues") tego komponentu będą

- przypisywane
- Edit — modyfikacje szczegółów komponentu
- Delete — usuwanie komponentu

Opcje domyślnego wybierania osoby przyporządkowanej do komponentu:

- Project Default — ten sam domyślnie przypisywany, co w projekcie powyżej
- Project Lead — będzie przypisane do "project Leader"
- Component Lead — będzie przypisane do "Component Leader"
- Unassigned — nie będzie przypisanie w momencie tworzenia sprawy

Gdy komponent zostanie utworzony w projekcie, we wszystkich sprawach ("issue") pojawia się dostępne pole "Component".

Konfigurowanie opcji JIRA

Kilkadziesiąt parametrów. Zmiana wymaga posiadania uprawnień 'JIRA System Administrator'.

Źródła

Amazon

<https://forums.developer.amazon.com/articles/193/what-do-user-permissions-mean.html>

Facebook

<https://developers.facebook.com/docs/facebook-login/permissions/>

Atlassian JIRA

<https://confluence.atlassian.com/adminjiraserver073/managing-project-permissions-861253293.html>

<https://confluence.atlassian.com/adminjiraserver073/managing-project-roles-861253314.html>

<https://confluence.atlassian.com/adminjiraserver073/managing-project-role-membership-861253319.html>

<https://confluence.atlassian.com/adminjiraserver073/creating-a-notification-scheme-861253478.html>

<https://confluence.atlassian.com/adminjiraserver073/configuring-issue-level-security-861253265.html>

<https://confluence.atlassian.com/adminjiraserver073/working-with-workflows-861253510.html>

<https://confluence.atlassian.com/adminjiraserver073/working-in-text-mode-861253635.html>

<https://confluence.atlassian.com/adminjiraserver073/defining-status-field-values-861253256.html>

<https://confluence.atlassian.com/adminjiraserver073/configuring-issues-861253230.html>

<https://confluence.atlassian.com/adminjiraserver073/configuring-workflow-schemes-861253569.html>

<https://confluence.atlassian.com/adminjiraserver073/defining-issue-type-field-values-861253235.html>

<https://confluence.atlassian.com/adminjiraserver073/associating-field-behavior-with-issue-types-861253410.html>

<https://confluence.atlassian.com/adminjiraserver073/managing-your-workflows-861253558.html>

<https://confluence.atlassian.com/adminjiraserver073/associating-issue-types-with-projects-861253240.html>

<https://confluence.atlassian.com/adminjiraserver073/defining-a-project-861253214.html>

<https://confluence.atlassian.com/adminjiraserver073/managing-global-permissions-861253290.html>

<https://confluence.atlassian.com/adminjiraserver073/managing-components-861253335.html>

<https://confluence.atlassian.com/adminjiraserver073/configuring-jira-application-options-861253962.html>