

Exporter issue per JIRA CLOUD

Exporter issue per JIRA CLOUD

In questo post andremo ad esaminare un addon molto importante per JIRA. Si tratta di *Exporter issue per JIRA CLOUD*.

 **ATLASSIAN**

Di cosa si occupa?

Fondamentalmente si occupa di esportare/importare le Issue dei nostri progetti presenti sulle istanze di JIRA CLOUD. In questo modo abbiamo un ulteriore strumento che aiuta nella nostra gestione ed amministrazione dei nostri progetti.

ATLASSIAN

Consente di eseguire delle esportazioni su formato CVS e

permette di poter successivamente reimportare il tutto sia sulle nostre istanze Cloud, che nelle nostre istanze Server. Questo diventa quindi un valido strumento per le **migrazioni** □



Il formato con cui viene eseguita l'esportazione è simile a quello che la stessa Atlassian mette a disposizione di JIRA. Quindi non abbiamo stravolgimenti che ci fanno ammattire, ma uno strumento che ci aiuta e ci mette a disposizione le informazioni esattamente come ci aspettiamo.



Possiamo anche vedere il **transaction log**, come mostrato in figura. L'esportazione risulta decisamente completa.

Considerazioni

Faccio alcune considerazioni sull'addon. Si tratta di un addon che colpisce nel punto giusto: le migrazioni e le esportazioni dati da e per i prodotti della Atlassian sono sempre stati una croce ed una delizia di ogni amministratore dei prodotti della Atlassian.

Avere a disposizione uno strumento completo che permette di poter, aggiungo io *Finalmente*, trattare l'argomento in maniera semplice e senza dover ammattire, è un vantaggio enorme.

Conclusioni

Abbiamo visionato un addon che ci aiuta nel nostro lavoro. Nei prossimi post andremo a fare le prove su campo di questo addon. Tenteremo di migrare un JIRA e ne valuteremo le potenzialità, limiti e prestazioni.

Reference

- [Video](#) che spiega il funzionamento dell'addon
- [Articolo di un blog](#) che presenta e spiega (in inglese) l'addon.

Product Requirements

Product Requirements

In questo post andremo ad esaminare come Confluence ci può aiutare nella redazione di product requirements, basandoci sull'articolo del [blog ufficiale di Confluence](#)



Andiamo al sodo

Sfruttiamo le funzionalità che Confluence ci mette a disposizione :-). Per realizzare il tutto abbiamo a disposizione opportuni template che ci aiutano nel creare questi contenuti particolari. Semplicemente andiamo a selezionare il template del **Product Requirements** e andiamo a costruire la nostra pagina.



Seguiamo l'autocomposizioni, cui la Atlassian ci ha abituati in tutti questi anni.

Let's get started

With product requirements you can...

- 1 Define document properties**
Add properties like status and owner to your document to make it easy to organise and sort your product requirements.
- 2 Create requirements**
The customisable template brings structure and consistency to your product requirements. If you use JIRA you can create stories right from the requirements page.
- 3 Track progress**
See the status of all your requirements at a glance. Sort by properties like status and release, or access linked JIRA issues from your requirements pages.

| | |
|-----------------|-----------------------------------|
| Target release | 1.0 |
| Epic | TIS-7 - Mobile IN PROGRESS |
| Document status | DRAFT |
| Document owner | @Ryan Lee |
| Designer | @Cassie Owens |
| Developers | @William Smith |

5 JIRA links

JIRA links

- Epics
Mobile

Don't show this again

Back **Create** Close

Come possiamo vedere, ci vengono fornite tutte le indicazioni del caso. Non possiamo sbagliare ☐

Quindi selezioniamo **Create**. Il risultato è il seguente


Titolo della pagina

Page properties

| | |
|-----------------|---|
| Target release | <i>Release name or number</i> |
| Epic | <i>Link to related JIRA epic or feature</i> |
| Document status | DRAFT |
| Document owner | Fabio Genovese [Administrator] |
| Designer | <i>Lead designer</i> |
| Developers | <i>Lead developer</i> |
| QA | <i>Lead tester</i> |

A questo punto compiliamo i vari campi e lasciamo libera la nostra fantasia, andando a scrivere tutte le informazioni che servono :-).

Che altro possiamo fare?

Ovviamente questo è il solo punto di partenza. Confluence ci mette a disposizione tutta una serie di strumenti che permettono di poter personalizzare in base a tutte le esigenze. Possiamo infatti costruire dei template ad hoc, espressamente dedicati per determinati Product Requirements, oppure avere delle diversificazioni in base ai progetti o alle tipologie di progetti. Oppure possiamo sfruttare altre caratteristiche, di cui abbiamo più che ampiamente parlato, quale il [Yoikee Creator](#), [Mind Mapping](#) 

In questo Confluence ci aiuta in questa operazione. Possiamo aggiungere altri addon, come descritto in altri [post](#), per poter estendere le funzionalità e aiutarci nel nostro lavoro.



Ripeto il consiglio: Lasciate che il vostro unico limite sia la fantasia :-D.

Conclusioni

Abbiamo visto un ulteriore esempio di come Confluence si dimostra uno strumento eccezionale, sempre rispondente alle nostre esigenze e con caratteristiche superiori.

Reference

- [Articolo del blog](#)

JQL – Esempi di uso

JQL – Esempi di uso

Vediamo in questo post alcuni esempi di uso del [JQL](#), ovvero del JIRA Query Language. Si tratta di alcuni esempi di utilizzo che possono aiutare nella vita di tutti i giorni ☐



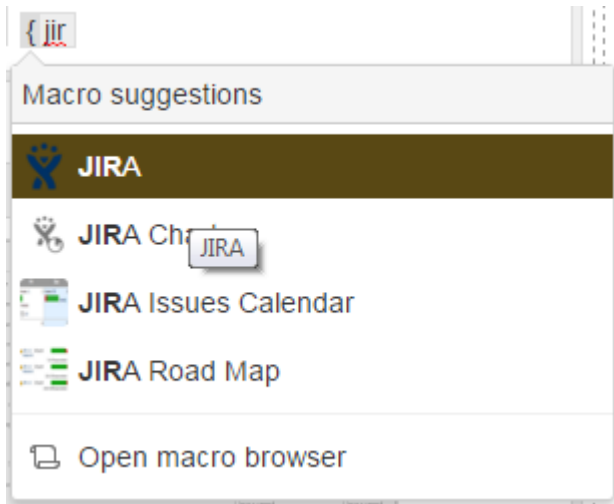
Andiamo al dunque

Dopo aver presentato il JQL, vediamo dove è possibile utilizzarlo per svolgere il nostro lavoro.

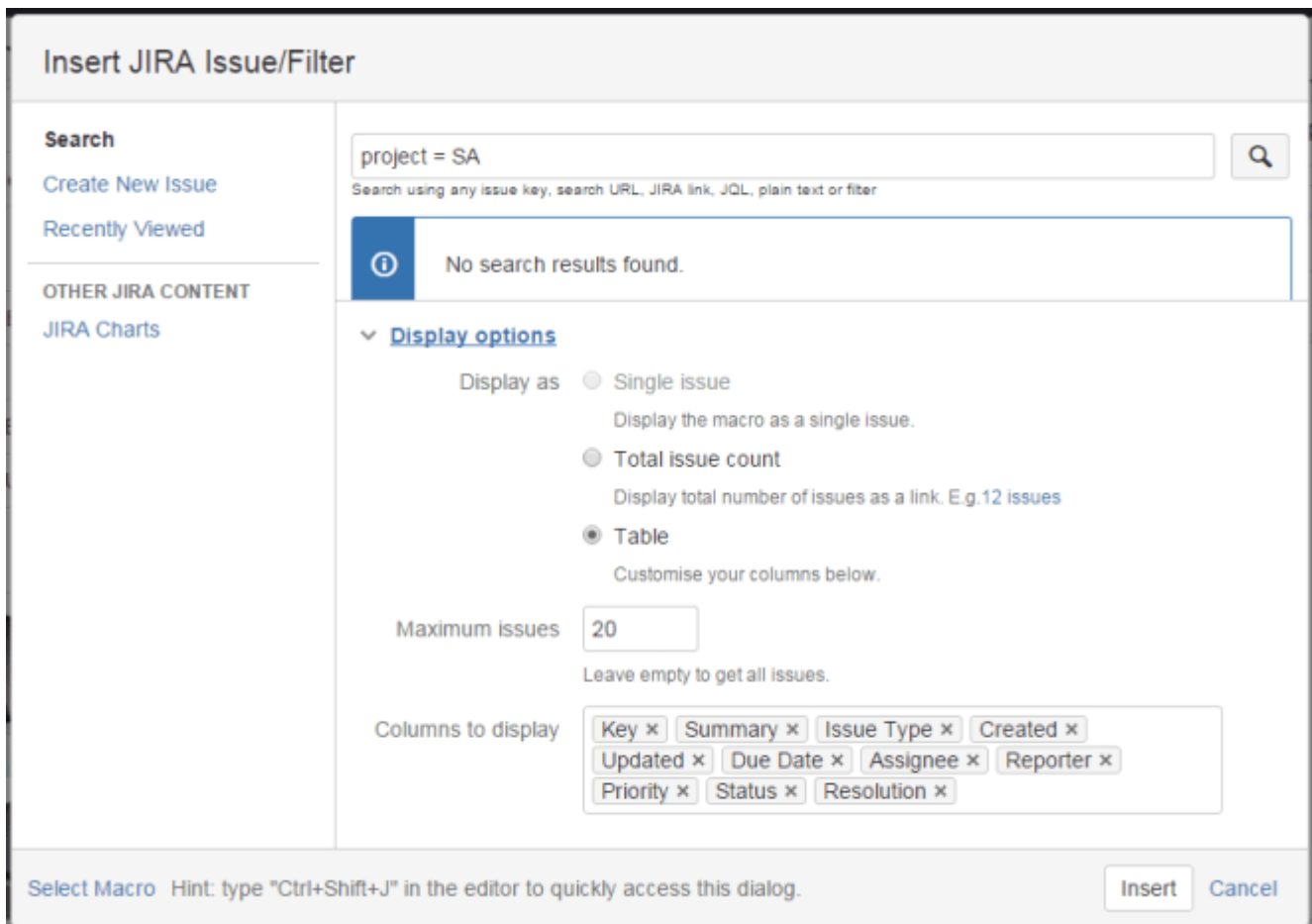


Confluence

Su Confluence abbiamo diverse possibilità di utilizzo. Abbiamo già mostrato alcuni di essi proprio nella gestione degli Asset Management, dove abbiamo anche collegato alle pagine dell'asset le Issue di riferimento attraverso una apposita macro: **JIRA**; come mostrato in figura:



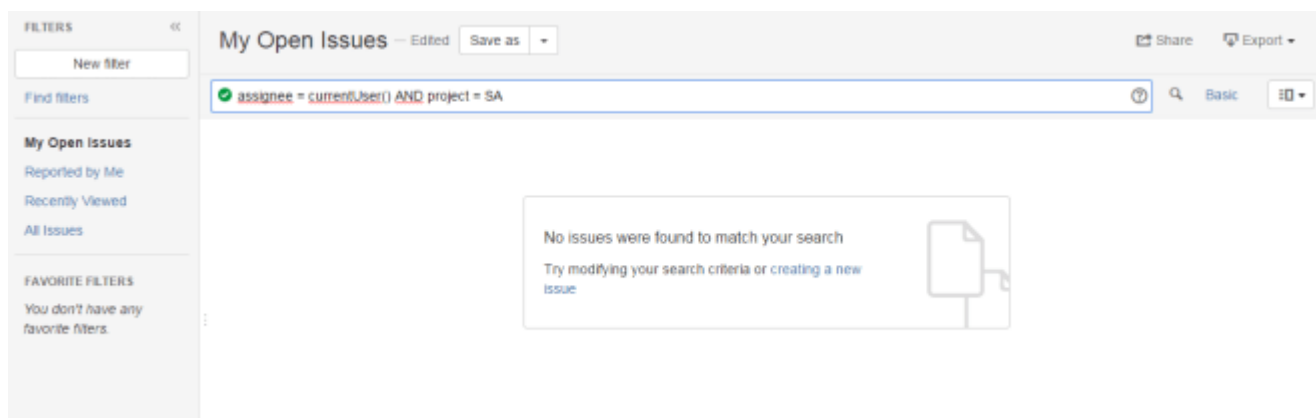
Quindi, sfruttiamo il JQL per andare a selezionare le Issue o la Issue che ci interessa.



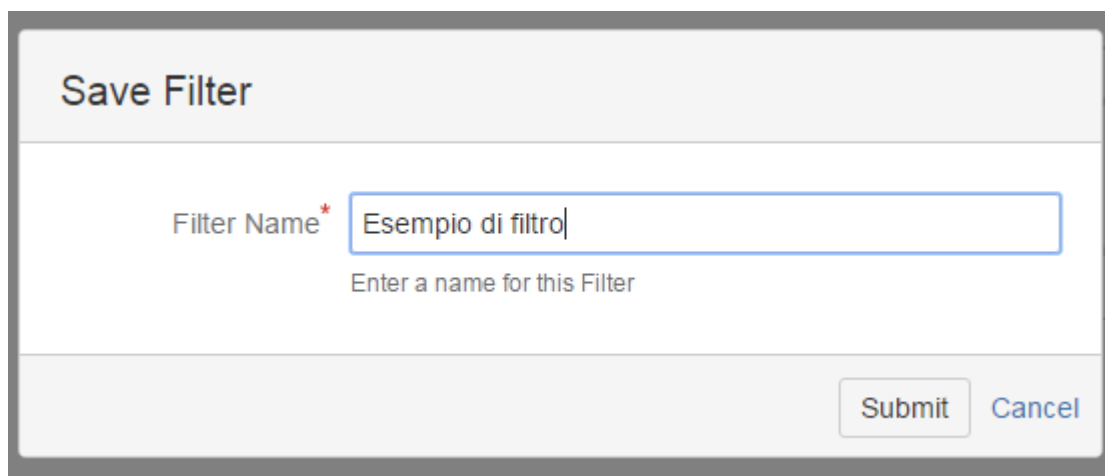
Come possiamo vedere, esaminando l'ultima immagine, quello che notiamo è che andiamo a selezionare le issue semplicemente andando a comporre la nostra query JQL in maniera opportuna. Quindi, senza fare alcuna fatica, settiamo le proprietà e le informazioni che vogliamo vedere, in modo quasi istantaneo □

JIRA

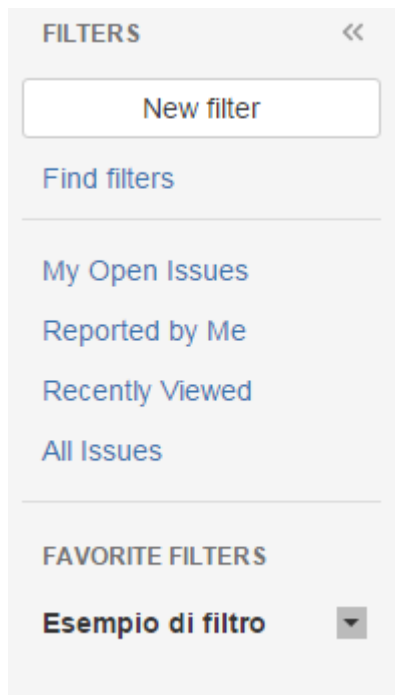
Su JIRA semplicemente andiamo ad accedere alla sezione delle Issue, dove andiamo ad impostare sia i filtri che ad eseguire tutte le interrogazioni che ci servono.



Una volta identificati i filtri che servono, possiamo semplicemente andare a salvare le query come dei filtri. Semplicemente selezioniamo il tasto **Save as** posto in alto (vedi precedente figura):



Una volta salvato, possiamo referenziare il filtro semplicemente dalla toolbar laterale:



Conclusioni

Abbiamo visto alcuni esempi di applicazione di JQL. Nei prossimi post andremo a vedere come possiamo estenderlo in maniera opportuna.

JIRA 7 a Bologna

JIRA 7 a Bologna

Il 27 Ottobre si è svolto a Bologna un evento dedicato a JIRA 7. Ho avuto la fortuna di poter partecipare all'evento e di poter vedere le ultime novità di JIRA 7. In questo post, e nei seguenti, cercherò di riassumere quanto è stato presentato in

questo evento □



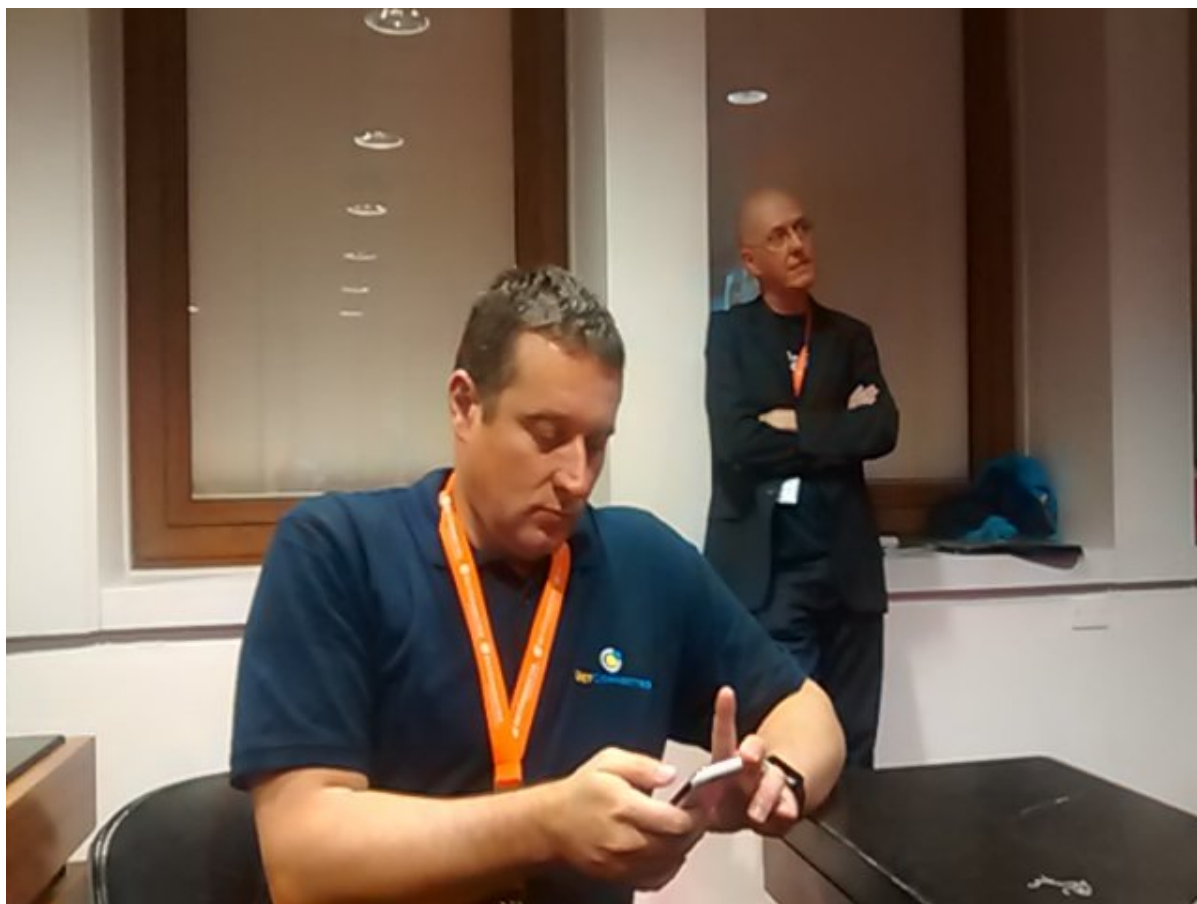
Team più performanti con JIRA

Il core della presentazione verteva sulla presentazione delle novità di ***JIRA 7***, presentate da un ospite di eccezione: ***Vladimir Cavalcanti***; EMEA Experts Manager Atlassian.



Dopo una breve introduzione, da parte di **Alessandro Rizzoli** di GetConnected, Vlad Cavalcanti ha iniziato una presentazione, tutta dedicata alla Atlassian, fornendo una panoramica sui vari prodotti e sui clienti (ben [50.000](#)), tra cui anche la

[NASA](#), dove la Atlassian ha fornito il suo contributo per la missione Rover Mars.



Vlad ha poi subito introdotto le ultime novità su JIRA 7, descrivendo le nuove pacchettizzazioni:

- [JIRA CORE](#)
- [JIRA SOFTWARE](#)
- JIRA Service Desk

Durante la presentazione ha subito evidenziato una delle domande più comuni, tra quelle che solitamente vengono poste durante le presentazioni: ***Gli Addons rimangono compatibili?*** La risposta: ***Rimangono compatibili con la nuova versione.***

Ha quindi evidenziato le indicazioni per le licenze: Nel caso dei vari prodotti, vince la licenza con il taglio più alto.

Vlad Cavalcanti ha quindi concluso il suo intervento mostrando un semplice caso d'uso di tutti i giorni, un esempio di situazione che si è presentata nel suo lavoro e di come, i

prodotti della Atlassian, lo hanno aiutato , soffermandosi come questi strumenti aiutano i team non IT



Si è quindi soffermato anche sulle ultime novità di Hipchat, segnalando che sarà presto rilasciata la versione Server per questo prodotto.



Ha concluso il suo intervento parlando degli [Ship-It Days in Atlassian](#), dove tutti aiutano con nuove idee. Ripeto: TUTTI. Una piccola curiosità: JIRA Service desk è nato a seguito di uno Ship-It :-).

Segue una presentazione, a cura di Cecilia Barbardi, sulla Continuous integration e sulla esperienza in azienda GetConnected.



Cecilia ha fornito una ottima spiegazione dei processi aziendali, sulla realtà e su come i prodotti della Atlassian, attraverso la loro integrazione, sono un valido supporto. :-). Passare da un sistema di scambio di file ad un sistema di integrazione, condivisione, scambio, ma soprattutto UNICO :-).

Questo è il valore aggiunto dei prodotti Atlassian.



Seguita una ottima presentazione di Luca, su come JIRA Service Desk aiuta i gruppi di lavoro nello svolgere il proprio lavoro, mettendo a disposizione sistemi semplici e veloci per la gestione delle anomalie e delle informazioni.



L'evento è stato chiuso da un intervento su **JIRA Portfolio**: Il prodotto più giovane della famiglia Atlassian. **Federico Sita** ha mostrato, con una demo molto completa e chiara, come questo prodotto può essere di aiuto nelle simulazioni e delle pianificazioni trasversali i vari gruppi di sviluppo.

Conclusioni

Un grande evento per Bologna, una occasione di potersi confrontare con esperti del settore e con varie esperienze sull'argomento. Sicuramente da ripetere

Un semplice esempio di Release Notes

Altro esempio di uso

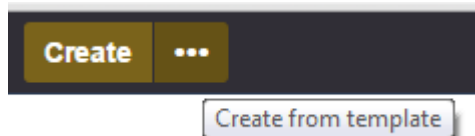
In questo post andremo ad esaminare quanto descritto nel post del blog ufficiale della Atlassian, cercando di spiegarlo meglio e di fornire quanti più dettagli possibili.



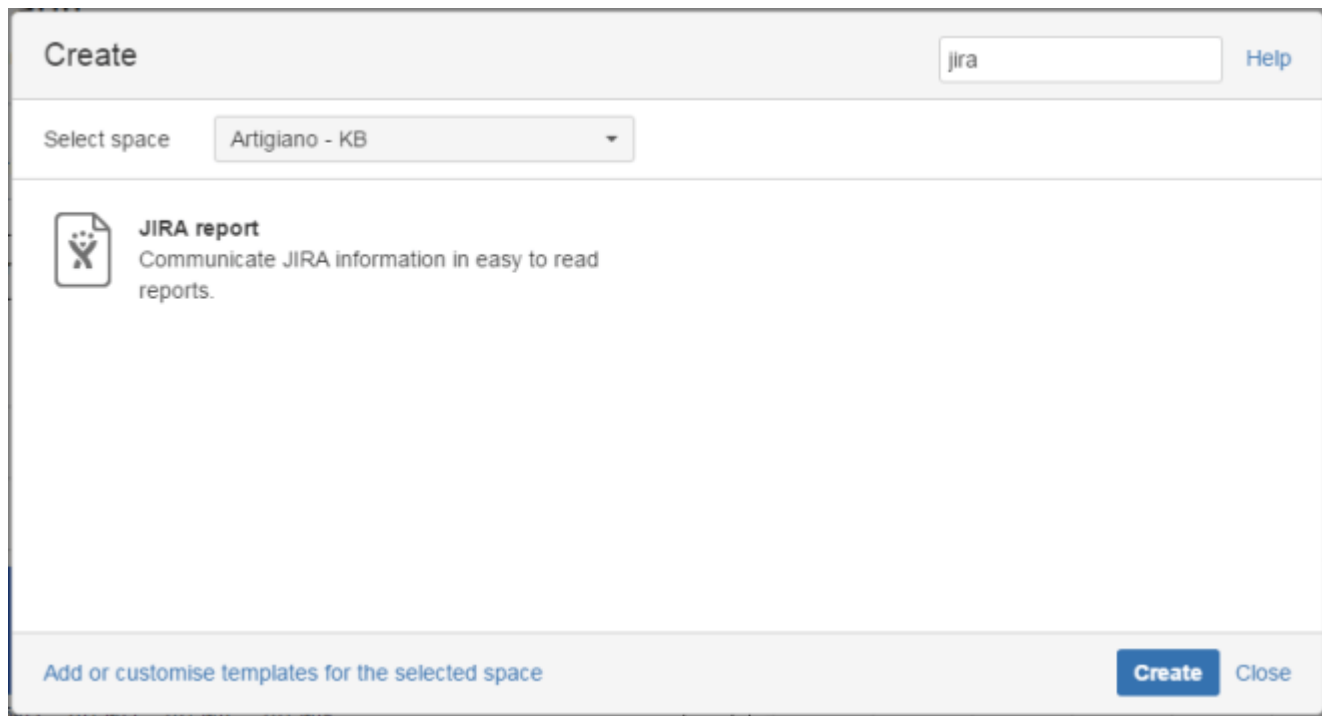
Procediamo...

Andiamo ad esaminare una funzionalità di Confluence, che ci consente di interfacciarci a JIRA, in modo da poter andare a leggere le informazioni delle issue, relative ad un rilascio, e pubblicare il risultato su di una pagina , che raccoglierà tutte queste informazioni.

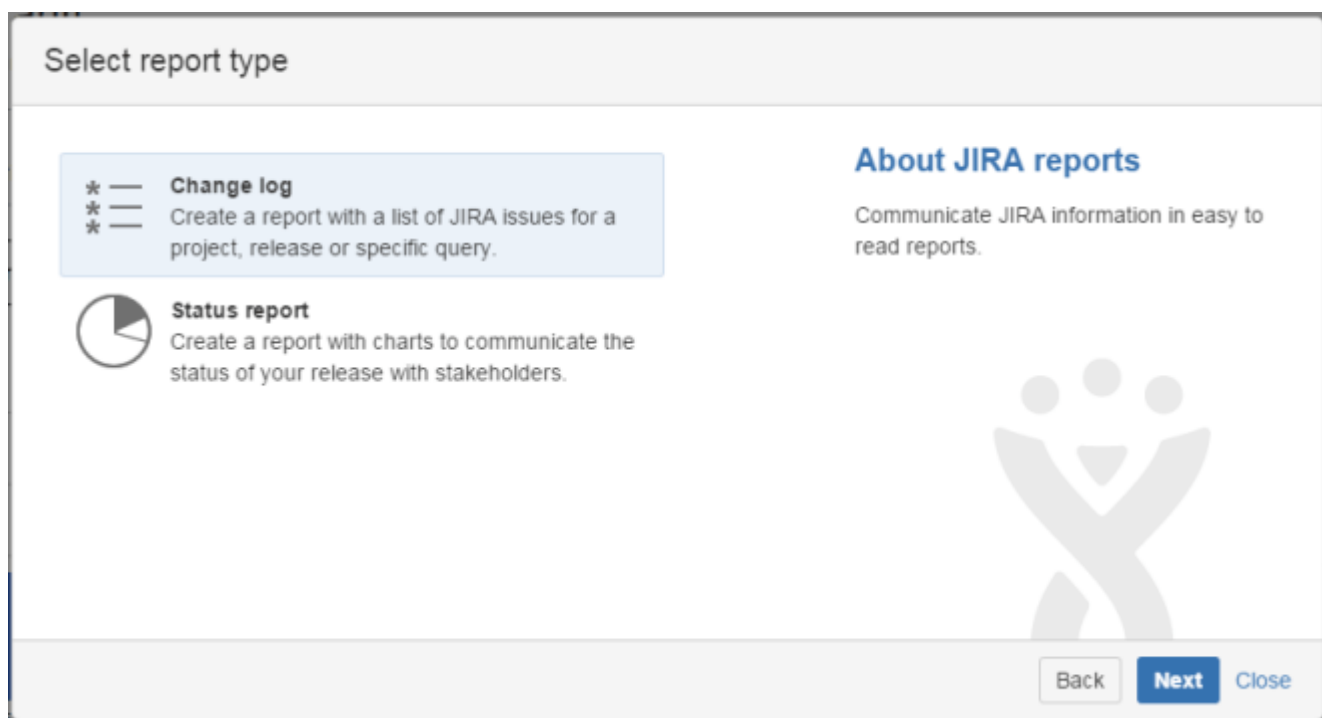
Andiamo a selezionare dal tasto:



e selezioniamo il template **JIRA report**.



quindi selezioniamo **Change log**:



A questo punto, attendiamo che l'autocomposizione di Confluence vada a leggere le informazioni dei progetti JIRA e ci chieda quali parametri passare:

Create a change log

Project


Fix version(s)

[Switch to advanced](#)

Title

About change logs

Keep a log of your teams progress or communicate deliverables. Generate a static or dynamic list of JIRA issues from a saved search, JIRA URL or JQL query.



Possiamo andare a imputare i parametri sfruttando la modalità semplice della autocomposizione, oppure, come mostrato nella seguente figura:

Create a change log

JIRA Query


[Switch to simple](#)

Title

Automatically update issue status and summaries from JIRA

About change logs

Keep a log of your teams progress or communicate deliverables. Generate a static or dynamic list of JIRA issues from a saved search, JIRA URL or JQL query.



possiamo impostare una opportuna query JQL, che ci consente di poter andare a selezionare le issue che ci servono :-). Il risultato è la seguente pagina:



Possiamo identificare le seguenti sezioni:

- **Page properties**, dove andiamo ad inserire dei metadati che possono risultare utili nel caso di report o recupero delle informazioni;
- **Riassunto**, dove riportiamo un riassunto delle informazioni;
- **Punti chiave del rilascio**, dove riportiamo che cosa si è voluto rilasciare
- **JIRA Issue**, dove è riportato l'elenco delle JIRA ISSUE coinvolte.

Come si vede, abbiamo a disposizione un semplice template che ci consente, attraverso l'uso di pochi passi e delle macro di connessione ai dati JIRA, di poter reperire le varie informazioni, in modo semplice e veloce.

Conclusioni

Abbiamo visto fino a qui delle indicazioni su come realizzare una Release Notes. Una cosa vorrei far notare: Questo funziona anche se non si tratta di un software, ma del rilascio di una nuova macchina, che presenta dei miglioramenti rispetto alla precedente edizione. Analogo discorso per un manuale. Ricordate quindi che quanto dico si può tranquillamente applicare anche ai progetti ed alle realtà NON IT, non di informatica :-).

Reference

- [Articolo del Blog della Atlassian.](#)
-

JQL – First look

JQL aka JIRA Query Language

In questo post andiamo ad esaminare questo semplice linguaggio, che consente di poter eseguire delle interrogazioni sulle ISSUE JIRA, come se si trattasse di usare una sintassi simile-SQL.

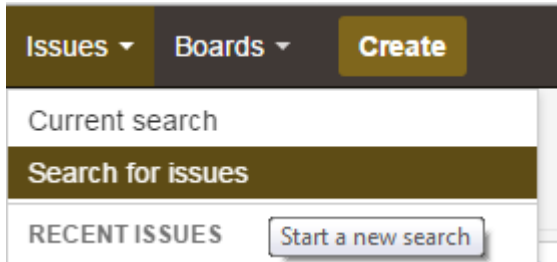


Che cosa è?

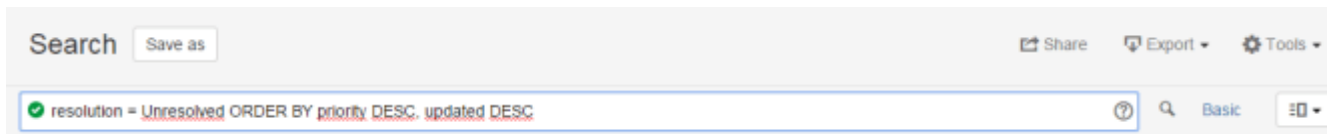
La prima cosa che facciamo è rispondere ad una semplice domanda: Che cosa è JQL? Come indicato nel titolo, JQL sta per JIRA Query Language ed è un linguaggio simil-sql che la Atlassian mette a disposizione per ricercare issue su JIRA.

JQL è stato studiato per TUTTI. Non è inteso come uno strumento per pochi, ma per tutti coloro che ne abbisognano. Consente di poter cercare facilmente ed agevolmente le Issue che interessano o abbisognano.

Possiamo agevolmente accedervi dal menù: **ISSUE** -> **Search for issue** come mostrato in figura.



Viene quindi proposta la form JIRA per le ricerche:



Dalla precedente immagine già abbiamo una prima indicazione di come si scrivono queste simil-sql query. In particolare:



Semplicemente andiamo a scrivere delle sequenze di:

- Campo
- operatore
- valore



JIRA ci aiuta con dei suggerimenti (vedi immagine precedente), fornendo dei suggerimenti nella composizione della query **JQL**. Se torniamo all'esempio della precedente immagine, notiamo che stiamo interrogando JIRA per capire quali sono le issue che risultano **NON RISOLTE**, ovvero **resolution = Unresolved**, ordinate per priorit  ed aggiornamento.

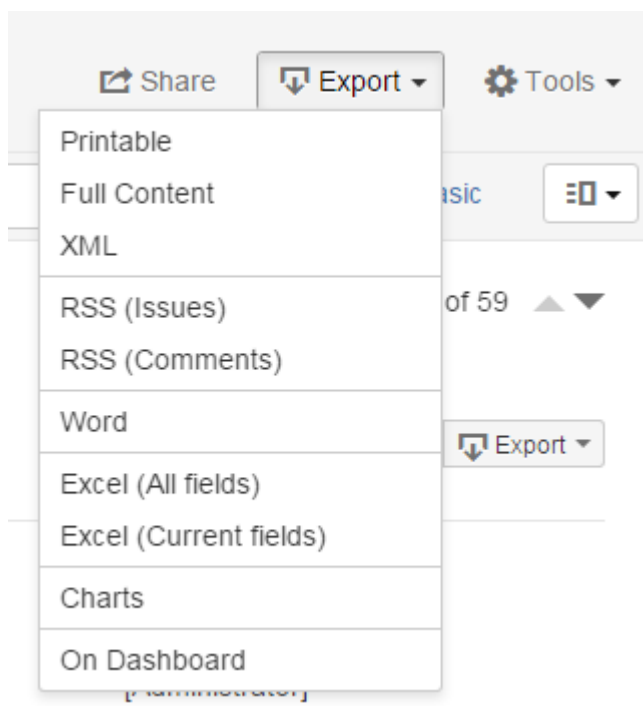
Notiamo anche che JIRA stesso ci indica se la query JQL   corretta o meno



aggiungendo una icona che indica se OK o meno. Il risultato è sicuramente ottimo.

Grande risultato, ma possiamo esportare i dati?

Certo che sì. La risposta è positiva. Una volta che abbiamo selezionato le informazioni che ci interessano, le issue che vogliamo, possiamo esportarle su Excel :-).



Attraverso l'apposito menù **Export**, visualizziamo le possibili alternative. Possiamo esportare tutte le informazioni che vogliamo. In questo modo possiamo eseguire ulteriori indagini sulle informazioni presenti in JIRA ☐

Conclusioni

Abbiamo iniziato a scoprire questa piccola perla. Si tratta di una funzionalità che consente di poter reperire issue da JIRA, in base alle nostre esigenze.

Reference

- [Manuale tecnico di JQL](#)
- [Articolo del blog ufficiale Atlassian](#)

Organizziamo una roadmap in Confluence

Organizziamo una roadmap

In questo post andremo ad esaminare come poter realizzare una [roadmap](#) sfruttando le potenzialità di Confluence. Riprendiamo quanto riportato nel [post ufficiale della Atlassian](#) e proviamo a sviluppare l'argomento.



Procediamo

Andiamo subito al dunque. L'obbiettivo è quello di sfruttare le funzioni, di cui dispone Confluence. La prima cosa da fare è quella di avere subito chiari i seguenti punti:

- Identifichiamo i punti cardini della nostra roadmap
- Inseriamo tutte le informazioni nelle pagine, organizzandole nel modo opportuno
- indirizziamo le persone verso le informazioni che le riguardano.

Possiamo iniziare a riassumere il tutto attraverso la seguente pagina:



Andiamo ad esaminarla ;-), analizzandola in ogni singolo punto.

Come prima cosa vediamo che le informazioni sono organizzate in due colonne, consentendo una migliore organizzazione della pagina. Un suggerimento che sempre ai vari utenti, è di fare un sapiente uso delle sezioni. Dalla versione 4 di Confluence, l'editor mette a disposizione la possibilità di poter organizzare varie sezioni già in colonne. A loro volta, è possibile aggiungere ulteriori sezioni (con il tempo e l'avanzare delle versioni, Confluence si è notevolmente evoluto □):

In questa pagina notiamo anche l'uso del componente standard Userprofile, che abbiamo ampiamente usato e spiegato in vari [post](#) e della [Roadmap](#), anche questa ampiamente spiegato :-). Infine, abbiamo le [Page Properties](#), che ci aiutano tantissimo.

La prima cosa che notiamo è che in questa pagina riusciamo a riassumere tantissime informazioni. Abbiamo subito le

indicazioni del gruppo di lavoro (lo Userprofile è uno strumento molto versatile ☐) ed anche il piano di battaglia relativo al lavoro da svolgere.



Questa è una delle pagine principali del progetto, dove abbiamo la prima impressione. E' una buona abitudine creare sempre una pagina principale che dia subito le informazioni principali. Poi da li, occorre inserire tutti i riferimenti alle altre informazioni.



L'uso della macro **Status** è utilissimo per arrivare ad identificare uno stato dello sviluppo, di un documento, di una procedura collegata o di qualsiasi altra indicazione relativa al progetto. Già in altri post, quali ad esempio dove abbiamo spiegato come realizzare una [scheda contribuente](#).



Aggiungiamo anche l'uso dei [task](#), che aiutano nella gestione dei compiti e, sfruttando la parte di messaggistica, consentono una rapida interazione con i vari membri del gruppo, anche se le persone sono distanti km e km.

Conclusioni

Abbiamo visto un esempio di come Confluence sia un valid strumento per aiutarci nel lavoro di tutti i giorni. Da evidenziare è il fatto che, nonostante sia nato per gli informatici, questo strumento ci aiuta notevolmente anche per progetti NON informatici.

Reference

- [Articolo del Blog Atlassian](#)
-

JIRA 7 – JIRA SOFTWARE

Continuiamo l'approfondimento

Proseguiamo il nostro viaggio verso le novità di JIRA. Iniziamo a parlare di JIRA SOFTWARE.



JIRA SOFTWARE

Come anticipato in questo post, abbiamo una nuova specializzazione di JIRA, espressamente dedicata per lo sviluppo software. Mentre abbiamo [JIRA CORE](#), dedicato ai progetti non IT, questa pacchettizzazione è espressamente pensata e rilasciata per il software.



JIRA SOFTWARE non è altro che JIRA, cui siamo già abituati, con già integrate le funzionalità di AGILE programming. Nulla di più. Il tutto espressamente dedicato allo sviluppo

software.



In aggiunta, sono presenti le ultime realizzazioni che, ad oggi, la Atlassian ci ha costantemente abituati, comprensive di migliorie che ci aiutano costantemente nella realizzazione del software. Segnaliamo anche ulteriori miglioramenti sulla grafica e sulla interfaccia web, come mostrato in figura.



Conclusioni

Abbiamo una nuova pacchettizzazione di JIRA che continua ad offrire molto agli sviluppatori. Con già inclusi i tools per lo sviluppo AGILE, ne fa uno tra i tool più potenti ☐

JIRA 7 – Ultime novità

Approfondiamo le ultime novità

In questo post andremo ad approfondire le ultime news su JIRA 7, già anticipate in questo [post](#), iniziando a capire quali sono le novità sostanziali.



Principali news

La principale novità è stata il ridisegnare JIRA in tre distinte pacchettizzazioni. Questo perché ci si è posti l'obiettivo di fornire uno strumento il più possibile calzante alle esigenze dei vari team di sviluppo.

Oggi parleremo di **JIRA CORE**, ovvero della pacchettizzazione dedicata a tutti i gruppi di lavoro che non si occupano principalmente di sviluppo IT, ma di altro genere di progetti.



Andiamo ad esaminare nel dettaglio quali sono le novità di **JIRA CORE**.

Sul [blog ufficiale della Atlassian](#), abbiamo questa indicazione:

What's JIRA Core?

JIRA Core is a project and task management solution that every business person across an organization can use to plan, track, and report on work. That means everyone using the same system. Cool, right? We've taken the fundamental features that business teams love and optimized them to fit their needs. This is great for JIRA admins, saving them time by reducing the need for customization work. JIRA Core's new business templates frees admins to more efficiently support multiple teams, projects, and workflows.

Fondamentalmente, dopo 13 anni di esperienza sul campo, la Atlassian si è accorta che molte aziende NON IT (ovvero che non sviluppano software), utilizzano JIRA per svolgere il proprio lavoro. Di conseguenza, hanno cercato di mettere loro a disposizione una versione di JIRA, da cui è stato estrapolato ciò che riguarda lo sviluppo software, mantenendo tutte le funzionalità base. Si è cercato, con questa

soluzione, di mettere a disposizione una versione di JIRA con le funzioni base, andando al sodo del lavoro.

L'obbiettivo è quello di mettere a disposizione un sistema di:

- Notifiche, con la possibilità di usare le **mentions** in modo da tracciare tutte le varie fasi e contattare direttamente le persone. Si sfrutta l'esperienza accumulata con Confluence[✖]
- Visualizzazione unica: Si mette a disposizione una visualizzazione unica per gestire le Issue.[✖]
- **Generalizzazione**: Abbiamo una versione più generica di JIRA. Si è cercato di dare uno strumento generico, non monodedicato allo sviluppo software. Personalmente la ritengo una scelta concettualmente importante, in quanto si mette a disposizione uno strumento che deve essere un aiuto, che disponga delle funzioni che servono e che non disponga di 1234567890987937352821632 di funzionalità di cui sono 3-4 funzionalità sono in uso. Possiamo usare JIRA CORE per aiutare le divisioni HR, legale, controllo di gestione, qualità, etc.[✖]
- Formalizzazione metodologia di lavoro: Sfruttando la potenza dei workflow, abbiamo la possibilità di poter formalizzare e rendere più semplice la metodologia di lavoro. Infatti, quello che possiamo fare è che creare dei Workflow per migliorare il lavoro, passando da [✖] ad un flusso più ordinato, chiaro e semplice da apprendere: [✖]Proviamo ad immaginare quali vantaggi possiamo trarre da questo strumento:
 - *Standardizzazione* del lavoro: non si improvvisa nulla.
 - *chiarezza*: ogni passo è chiaro e perfettamente definito.
 - *documentazione*: si può creare una opportuna documentazione di supporto che aiuta

nell'apprendimento dei nuovi
dipendenti/collaboratori;

Conclusione

Abbiamo iniziato a vedere le ultime novità sulla parte JIRA. Nei prossimi post andremo a visionare meglio anche le altre pacchettizzazioni, evidenziandone le differenze e i campi di applicazioni. Quindi andremo a provarle sul campo, come sempre, per saggiarne potenzialità e limiti.

Reference

Si consiglia il seguente video:

http://www.youtube.com/watch?v=cj9_wFQBDf4

Prendiamo una decisione...

Prendiamo una decisione ...

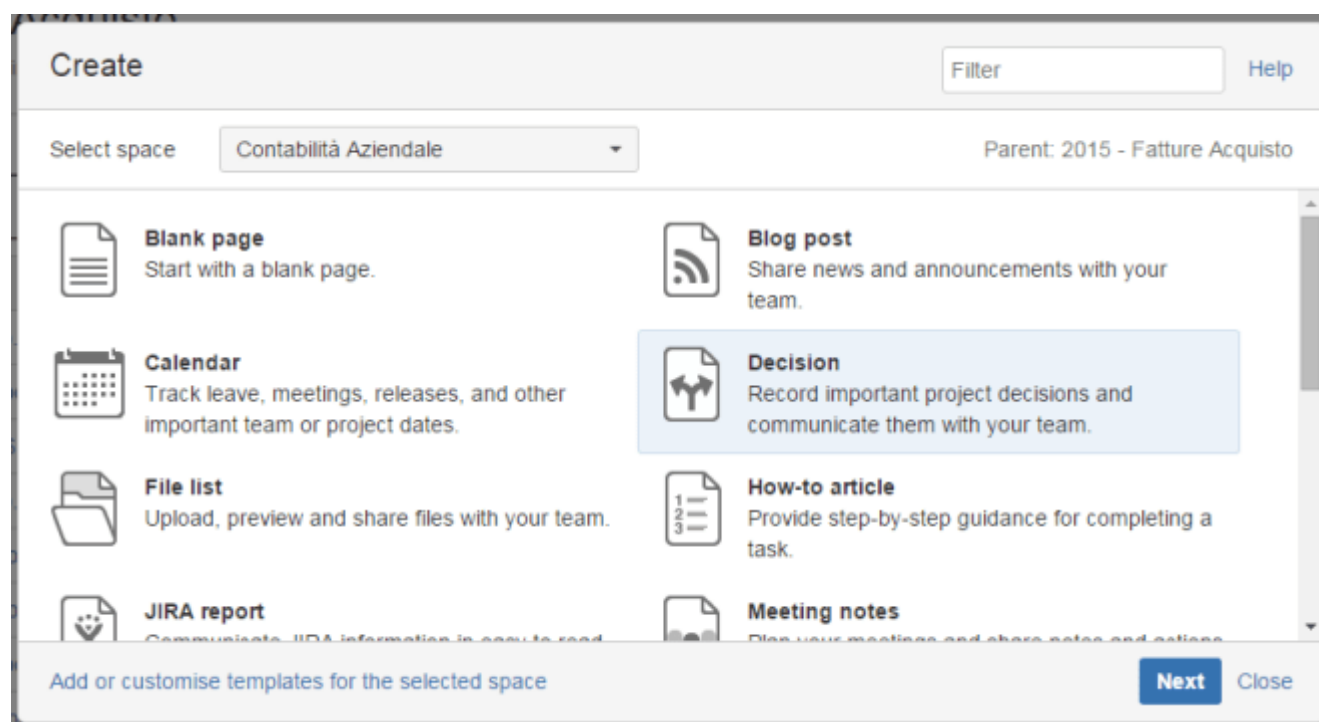
Sulla falsariga di una pubblicità, usiamo Confluence per aiutare gli utenti di un determinato gruppo di lavoro, nell'ambito di un loro processo decisionale, a prendere una decisione. Ci rifacciamo all'articolo del blog ufficiale della

[Atlassian](#), ma cercherò di metterci del mio :-P. L'obiettivo è quello di stimolare la fantasia degli utenti in modo che siano loro a costruirsi la propria soluzione ☐



Andiamo in dettaglio

Confluence ci mette a disposizione tutta una serie di strumenti che possono essere usati per definire un processo decisionale. Possiamo utilizzare sia i template che Confluence stesso mette a disposizione. Si tratta del template **Decision**, che come vediamo possiamo così usare:




Come possiamo vedere, inseriamo pochi semplici parametri, e arriviamo a costruire la nostra pagina dedicata alla decisione.

Track a decision

| | |
|--------------|--|
| Status | <input type="text" value="Not started"/> |
| Decision* | <input type="text" value="What are you deciding?"/> |
| Owner | <input type="text"/> |
| Stakeholders | <input type="text" value="Who needs to help make this decision?"/> |
| Due date | <input type="text"/> |
| Background | <input type="text" value="What details are important in making this decision?"/> |

About decisions

Work with stakeholders to decide important things, such as project scope or schedule changes. Share and track your team's decisions in a central register.



Tuttavia, questo non è una regola invalicabile. Possiamo usare anche altri [template](#), come mostrato in precedenza; possiamo anche costruirci il nostro template ad hoc.

Un esempio potrebbe essere questo che segue:

Artigiano del Software

Page properties | label


| | |
|--------------|--------------------------------|
| Status | IN PROGRESS |
| Stakeholders | Fabio Genovese |
| Outcome | What did you decide? |
| Due date | 22 Oct 2015 |
| Owner | Fabio Genovese [Administrator] |

Edit Remove

esempio di decisione

Possiamo usare il template che Confluence mette a disposizione, per poi estenderlo a nostro piacere e necessità, aggiungendo ulteriori componenti o togliendone di non necessarie. :-). Questo è il bello di Confluence: Libertà massima.



In questo caso, abbiamo usato i seguenti componenti:

- **Metions**, per coinvolgere gli utenti di confluence responsabili della ... decisione. Nel caso di altri utenti, non presenti in Confluence, possiamo solo indicare il nome. Questa soluzione è usabile anche nel caso di pagina che consente un accesso anonimo; 
- **Page Properties**, che andiamo ad usare per inserire una serie di informazioni o metadati alla pagina. Questo risulta assai utile quando vogliamo creare delle pagine

Esempio di una decisione

| Page properties label | |
|-------------------------|--------------------------------|
| Status | NOT STARTE... |
| Stakeholders | Fabio Genovese [Administrator] |
| Outcome | What did you decide? |
| Due date | 29 Oct 2015 |
| Owner | Fabio Genovese |

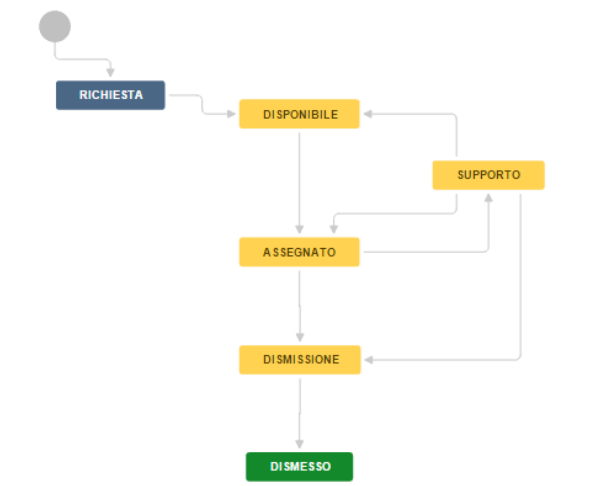
riassuntive

- **Table**, che usiamo per definire step, informazioni in maniera strutturata, etc. 
- **Messaging**, che andiamo ad usare per comunicare informazioni/messaggi/task. 

Carino, ma è il solo modo?

La domanda sorge spontanea. Possiamo solo usare Confluence? Azzardiamo una soluzione alternativa. Possiamo anche fare uso di JIRA, sfruttando le sue potenzialità. L'uso di JIRA può fornire quella marcia in più per poter aiutare le persone nel percorso decisionale.

Infatti, possiamo sfruttare i workflow in modo da impostare un percorso decisionale, definendo tutti i passi da seguire per arrivare alla decisione.



Questo ci può aiutare anche nel definire dei processi approvativi, come mostrato nell'esempio dell'[Asset management](#) e l'uso delle board Agile, ci può aiutare nella fase di cambio stato o nel mostrare lo stato di avanzamento (ovviamente in situazioni limitate o con opportuni filtri – filtri da verificare).



Usando JIRA, invece di creare delle pagine, andiamo a creare un progetto o più progetti, dedicati ai processi decisionali. Le relative Issue saranno poi le singole decisioni da prendere.

Possiamo sfruttare le potenzialità delle linked issue per legare eventuali decisioni ad altre, qualora ci siano delle dipendenze. Non male ☐

Possiamo anche aggiungere una serie di campi custom, dedicati alla fase decisionale, che sicuramente ci aiutano. Ovviamente, in questa soluzione, Confluence può essere o meno usato come supporto. Mi spiego meglio: possiamo usare sempre confluence per documentare / creare le pagine di documentazione che saranno di supporto. Ovviamente queste pagine possono anche essere i contenitori di allegati di vario genere e natura.



L'obiettivo è di sfruttare il più possibile le funzionalità che abbiamo a disposizione. □

Un piccolo chiarimento

Anche se i prodotti sono dedicati prevalentemente allo sviluppo software, possiamo usarli anche per soluzioni che definisco alternative. Quello che voglio esprimere in questi post, è che risulta possibile utilizzare questi strumenti anche per altri scopi, che nulla hanno a che fare con la IT.



Conclusioni

Abbiamo visto due possibili utilizzi delle funzionalità di Confluence e JIRA. Possiamo usare Confluence per aiutarci nel nostro processo decisionale, oppure usarli entrambi in modo da sfruttare meglio le funzionalità di entrambi. Ricordate sempre: La fantasia deve essere il nostro unico limite □

<http://blogs.atlassian.com/2015/09/make-better-decisions-software-team/>