

**MAKING  
WORKFLOW  
WORK  
FOR YOU**



COUNTERSOFT

## Contents

Making Workflow Work For You.....	2
What we're going to cover .....	2
Configuring Gemini Workflow .....	2
Item Types.....	2
Statuses.....	3
Status Transitions.....	4
Automated Transitions .....	5
Workflow and Workspaces .....	5
Notifications and reporting of status changes.....	5
Metrics and management.....	6
Workflow wrap-up .....	6

## Making Workflow Work For You

### What we're going to cover

Workflow in Gemini broadly translates to the change of status of an item from the point at which it is created in the system until it is closed. Making Workflow work for you describes the functionality in a way that might help you map your business processes onto Gemini's Workflow features. Here we shall cover:

- Workflow configuration
  - Item Types
  - Defining statuses
  - Defining status transitions
- The setting of the initial status
- The transitions from one status to another
  - Manual transitions
  - Automatic transitions
  - App-driven transitions
- Workflow and Workspaces
  - Notifications and reporting of status changes
  - Metrics and management

The metadata we will discuss here – Types, Statuses and Workflow – belong to a Project Template and if you are not familiar with project Templates then you might benefit from reading the documentation on them [here](#) before you go any further.

## Configuring Gemini Workflow

### Item Types

Workflow in Gemini is Type specific. A Type is a specific type of work item, for example a Change Request, Ticket, Bug, Test Case etc. Each of these can capture different data and have different field access permissions. Different Types on the same Project Template can share Workflow but each Type can have its own unique Workflow.

**Project Template:** Agile Development [Duplicate](#) [Export](#) [Load](#) [Delete](#)

» Add and maintain the different types of **Process** that projects based on this template will handle (e.g Task, Ticket, Issue). Specify data capture and field permissions on **Screens** and status changes and permissions on **Workflow**.

Add		Search:		
Type	Color			
Story		<a href="#">Screens</a>	<a href="#">Workflow</a>	
Task		<a href="#">Screens</a>	<a href="#">Workflow</a>	 
Change Request		<a href="#">Screens</a>	<a href="#">Workflow</a>	 
Bug		<a href="#">Screens</a>	<a href="#">Workflow</a>	 

Showing 1 to 4 of 4 entries

Fig 1.0 Types on a Project Template

## Statuses

Since Workflow belongs to Types these must exist before Workflow is defined. What must also exist in order to define Workflow are the Status values that mark the transition of work items from open to closed. We are not going to explain how to create or maintain Status vales on a Project Template, the documentation for that can be found [here](#).

**Project Template:** Agile Development [Duplicate](#) [Export](#) [Load](#) [Delete](#)

» Define the **Status** values for this project template. Also define **Workflow** and set user permissions in **Process > Workflow** after you have created the **Status** values. **Status** values are Template specific.

Add		Search: <input type="text"/>			
	Status	Comment	Color	Final?	
	In Backlog				
	In Sprint				
	In Progress				
	Tested				
	Closed			Yes	

Showing 1 to 5 of 5 entries

Fig 1.1 Status values on a Project Template

Once statuses have been defined, to configure workflow, simply click on the Workflow hyperlink for a specific Type (see fig 1.0). Each Project Template defines the full set of statuses that all the different Types of work item in it can pass through. Gemini will show a screen where the statuses are portrayed in a drawing area in boxes. Within each box is a colored square and by clicking and holding the mouse down over this square the user can draw arrows linking the status boxes, which themselves can be dragged and arranged on the canvas.

**Exercise:** log into Gemini (if you don't have a Gemini instance then you can take a free hosted trial), navigate to the Project Templates (you must be a Gemini Administrator to do this) by selecting Customize...Templates (it is the default starting point of the Customize section). In the drop-down list of sample Templates select the Helpdesk Template. You will see it defaults different Types – Ticket, Bug, Task etc.

Click on the Workflow hyperlink associated with Type Ticket. What you see is the list of statuses that an item of Type Ticket can go through, drawn from the total list of user-definable Statuses. Workflow defines the paths that an item can go through from the point at which it is entered into the system to the point at which it is closed. You do not have to use all of the statuses that exist on the Template against any individual Type.

Click on the arrow that marks the transition from one status to another and you will see that you can delete it (and change the Workflow); you will also see a User Group selector. You can define which groups of users are allowed to change the status of an item and from which values.

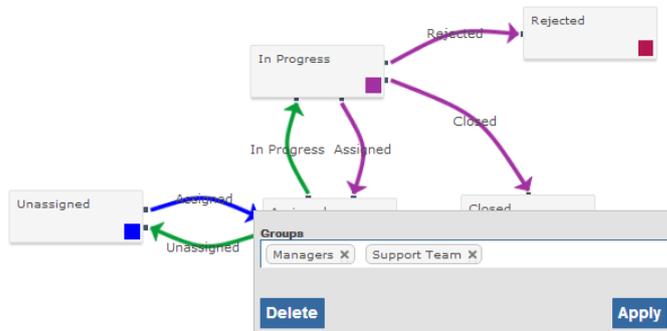


Fig 2.0 Sample Workflow showing a status transition being restricted to certain User Groups.

### Status Transitions

As the diagram above illustrates, you define the paths that a work item can go through with simple drag-draw actions, and on each transition you can further define which groups of users are allowed to make that transition.

### Manual Transitions

The simplest change of status occurs when a user with edit permissions on the Status field, clicks on the Status field and uses Gemini’s inline editing to change its value. When this happens, the list of available statuses, whether they represent forward or backward movement in the flow, are displayed.

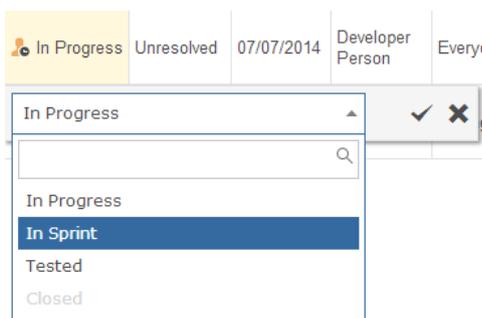


Fig 3.0 Manually changing status.

One thing worth noting is that the rule of [Dependencies](#) in Gemini is always enforced. In figure 3.0 above the Status ‘Closed’ is greyed out and not selectable. This is because the task in question has child tasks and the rule of Dependencies in Workflow is that a parent task cannot be closed unless all of its child tasks are first closed. This is one of the fundamental differences between Dependencies and [Linked items](#).

The first sentence in this section stated that the user must have Edit permissions on the Status field, so why then does Gemini have additional security on Status transitions? Two reasons: first the user might be able to edit the status but not at all stages in the Workflow, for example a user might be

able to Assign and task but not to Approve it. The second reason is that not all forms of editing are obvious, for example if the [Planning Board](#) axis is set by Status then the act of dragging an item from one status column to another would automatically result in a change of Status, so Gemini checks if the user concerned is able to make such a status change and if not it does not allow it.

### **Automated Transitions**

Status changes can be made to occur automatically. There are two ways to make this happen:

1. Through the app framework
2. Using Rules and Actions

#### ***Using the app framework***

One of the standard apps that is delivered with Gemini is [Assigned Resource Change Status](#). This app is for teams where the second status of an item, of any Type, is Assigned. This is an Event app and fires when the Resource field is populated, at which point it will automatically move the status on from its current value to the second value unless the status is already at or beyond that point. For example if the statuses were Unassigned...Assigned...In Progress...Closed and the item was in a status Unassigned then populating the Resource field would result in the status being automatically moved to Assigned.

As this is an Open Source app, like most Gemini apps, you can take the code and change its behavior or clone the app and create your own version. Event apps are not the only way to change Workflow programmatically, there are also Timer apps and UI apps that you can write, full code samples of which can be found at <http://countersoft.github.io>.

#### ***Using Rules and Actions***

Rules and actions allow for the change of status of an item non-programmatically. One simply defines the conditions that should trigger the rule and specify the status change as the action. Gemini will validate the user permissions, that the status transition path exists and the rule of Dependencies before actioning the change.

For more information on Rules and Actions and to see a brief demonstration of this feature in action please see the documentation [here](#).

### **Workflow and Workspaces**

Workflow is often encapsulated by Workspaces. For a full list of the functionality underpinning Gemini's Workspaces see [<hyperlink this> Getting Productive with Workspaces](#).

#### **Notifications and reporting of status changes**

Workspaces define their scope by virtue of the cross-project filter, which allows work items from any project and in any status to be made subject to a Workspace. Many users will therefore create Workspaces that are based in filters, which include the Status field. Simple examples of this are statuses that indicate managerial approval is needed or that a piece of work has moved from the scope of one team to another – a dev team setting the status of an item to 'Ready for Test' for example.

Because of their in-built alerts and alert subscription model, scheduled reporting, and badge count notifications, Workspaces are the perfect way to keep all stakeholders aware of status changes and Workflow progression.

### Metrics and management

Sophisticated metrics are normally the province of the [Progress](#) menu, or advanced [Excel reporting](#), however Workspaces contain [instant metrics](#) that are extremely useful to many teams. Metrics such as Open Items Summary, Burndown rate, and various data constituency charts are instantly available to the team.

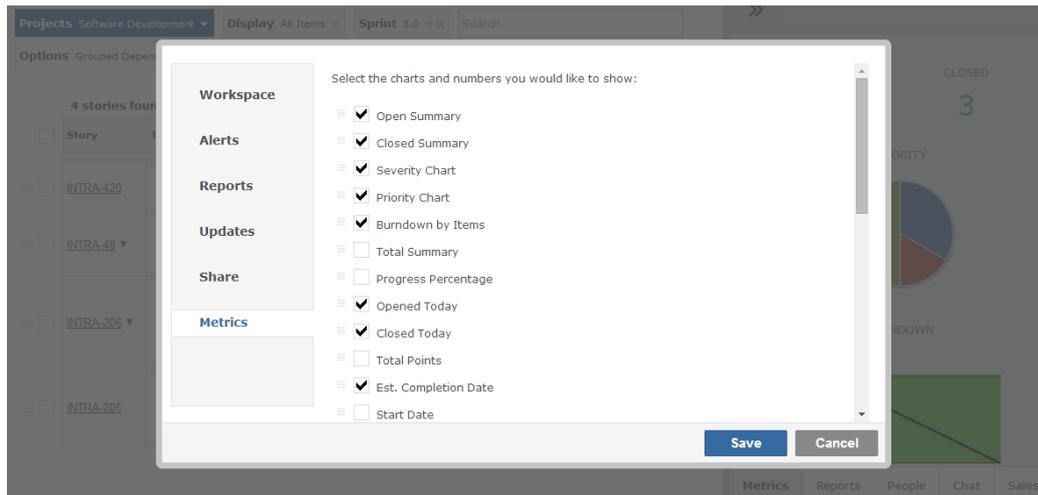


Fig 4.0 Metrics from the toolbar give instant insight into the work in a Workspace

### Workflow wrap-up

It seems only fitting that having worked out way through Workflow we should draw some conclusions at the end. Gemini gives you the ability to:

- Define Workflow with simple drag-drop on a drawing surface
- Secure Workflow at a granular level by specifying who can change what and when.
- Implement non-programmatic rules-based status updates.
- Programmatically update statuses with an apps framework that will even let you integrate with external systems to drive Workflow (did the DevOps script execute to deploy successfully, if so set Status to 'Deployed' etc.).
- Automatically draw the appropriate tasks into the scope of your Workspaces, which provide instant visibility, scheduled reporting, collaborative capability and more.

Making Workflow work for you requires nothing more.