

# ESSENTIAL BUG & ISSUE TRACKING



COUNTERSOFT

## Contents

Essential Bug & Issue Tracking .....	2
What we're going to cover .....	2
Tools, Integrations and Extensibility .....	3
Tools and Integrations .....	3
Extensibility .....	3
Supporting Scrum/Agile .....	4
Scrum .....	4
Agile .....	5
Flexible Workflow, Rules & Actions .....	7
Workflow .....	7
Rules & Actions .....	8
Handling stakeholder jargon – Template Taxonomy .....	10
The Workspace override .....	10
Smart Data Capture .....	10
Standard fields and built-in behaviours .....	10
Field level permissions .....	11
Custom Fields .....	11
Alternatives to manual data input .....	11
Security and Scalability .....	12
Security .....	12
Scalability .....	13
Gemini's Technical Architecture .....	14
One of a kind .....	14
Get your hands on the code .....	14
UI, UX and Reporting .....	15
The best UI/UX .....	15
Simple, powerful ad hoc and standard reporting .....	15
A complete package .....	15

## Essential Bug & Issue Tracking

### What we're going to cover

ALM-lite tracking applications and Scrum/Agile planning boards are fine for small teams developing software for users who initially are at arms length. Eventually however, those same users will seek to influence the product roadmap. Noise builds and balls get dropped. Complaints emerge from people who have little or no visibility of processes that affect *their* job function. *How does the sales team react to the customer who complains that their helpdesk ticket has not been responded to?* Internal stakeholders, at varying levels of authority and with differing information needs, demand that they too are brought into the loop, often bowing to external pressure to ensure that the customer has a voice as well.

At this point the early stage tracking tools break down. Some IT teams make the mistake of pushing their ALM tool out to the user base. *It works for their communication so why not for everybody else?* Users with little or no knowledge of tech jargon are expected to understand Sprints, Backlogs and Regression Testing. On top of the problems that they care about they have a new one – adoption of an alien tool to log and track their issues in.

If you recognize the scenario above you will understand why Gemini's approach to bug and issue tracking is to integrate IT functions that manage and resolve problems with the non-IT stakeholders who find and report them.

Here are our top 10 requirements for a bug and issue tracker suitable for IT and end-user audiences:

1. It must integrate with the tools that the delivery team use, with a framework of extensibility so that where necessary it can be fully customized to meet the team's needs.
2. It must support modern methodologies such as Scrum/Agile and yet be equally useful to teams that employ traditional Waterfall or custom methodologies because of the business environments in which they operate.
3. It must be flexible enough to adapt to the working practices of both the IT team and the non-IT stakeholders with variable Workflow and user-definable rules and actions.
4. For ease and breadth of adoption it must handle the jargon of multiple stakeholder groups.
5. It must have smart data capture, treating custom fields as if they came out-of-the-box.
6. It must take data input from any and all of the following sources:
  - a. Manual input from a web browser
  - b. Email (POP/IMAP and Exchange)
  - c. 3<sup>rd</sup> party systems integration through an API
  - d. Extensibility through an application framework
7. It must be deployable in secure environments and scalable so that it can be used by small teams or in a high availability environment where must support thousands of users.
8. It should utilize modern tools and architecture for the benefit of its users and the IT organization that have to support it.
9. It must have a slick UI, superb UX and flexible reporting. If it doesn't look good and you can't get data out of it easily, nobody will like it no matter what it does.

10. It must be part of a total quality package – well supported, documented, described in videos and honed by a discerning customer base through simple and complex scenarios.

## Tools, Integrations and Extensibility

### Tools and Integrations

Gemini can track issues in any kind of project, but bug and issue tracking is predominantly concerned with software development. To that end it has integrations with the most commonly used tools in the ALM space, including:

- Visual Studio
- Microsoft Team Foundation Server
- Git
- Mercurial
- SVN
- Testuff

Gemini's Workspaces have in-built team chat but it also has integrations with:

- Slack (used internally by the Gemini dev team)
- Flowdock

Office Automation and email integrations:

- Exchange and any POP/IMAP email
- Outlook
- Excel

Other integrations include:

- Geckoboard
  - LucidChart
  - MS Project
- more...

### Extensibility

#### *Apps framework*

Gemini has an apps framework. To keep the core of the product compact, robust and scalable, functionality that might otherwise bloat the codebase is delivered through the app framework. Roadmap, Changelog, DocStore, Code Review (Saucery), AD Integration, Test Plans and Test Cases (Sentry), and much more of Gemini's feature set, are all apps that users can choose to deploy. The majority of apps are Open Source so that customers can plug in their own functionality, contribute enhancements to the Gemini community, and run a lean codebase. The source code for all Open Source Timer, Event and UI apps is located at <http://countersoft.github.io>. For more information on developing Gemini event apps see the documentation [here](#).

## RESTful API

Gemini has a comprehensive RESTful API that is fully documented with sample code. For more information please see the online API documentation [here](#).

## Supporting Scrum/Agile

### Scrum

Gemini has one of the most sophisticated scrum boards in the world, with a host of unique features, such as:

- Independent swimlane scrolling
- Lane limits, card limits
- Dual axes
- Dynamic Rows
- Color coding of attributes  
more...

Because the board is a standard Gemini view it is reached through and a component of Workspaces. Workspaces are discussed in detail in the downloadable pdf *Getting Productive With Workspaces* **<THIS SHOULD BE HYPERLINKED>**. Workspaces provide a dynamic, cross-project filter of data, user-selected metrics, auto notifications and alerts, ad hoc and scheduled reporting and an instant view of user workload.

For more information on the Gemini Board see the documentation [here](#).

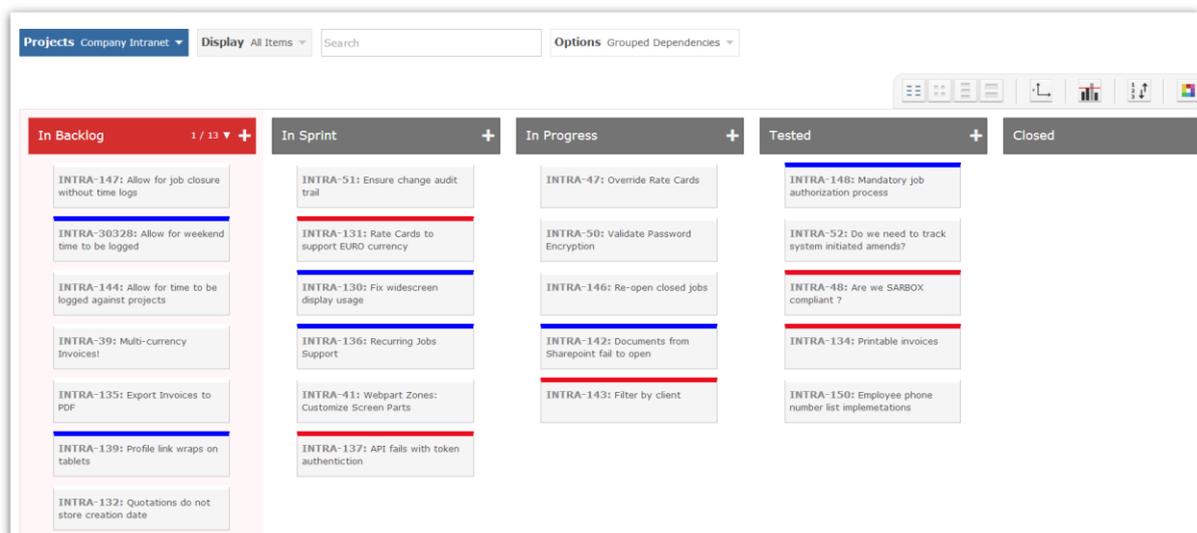


Fig 1.0 A snapshot of the Gemini Planning Board

**Note:** The filter that runs across the top allows item selection, first column shows 1/13 rows and allows swimlane to scroll, controls at top right handle 4 x zoom levels, dual axes with instant axis flip, lane limits, ordering and color coding of cards (note red/blue/grey as cards are colored by priority).

## Agile

### *Graphical data representation*

Gemini lets you set up sprints on a per-project basis and the start/end dates of each sprint are pulled through to the Burn-down and Burn-up charts. In addition to Burn-down and Burn-up charts, Gemini also plots Velocity (the rate at which a team has historically closed out tasks).

### *Estimating effort*

You can estimate time in man-hours of effort or in points and Burn-down and Burn-up charts can represent either or both units as well as being able to plot on daily, weekly or monthly basis.

### *Tracking time spent and estimating progress*

With simple point-click you can update the percentage of progress, which Gemini displays on various screens in a graphical manner. Every work item tracks the time that is logged against it (with time reason codes) and calculates the time remaining to complete based on any estimated hours. Gemini has an app extension that can be used to track the original hours that are estimated against a task so that if users have scenarios where estimates change but need to track the original values then those are never lost and can always be seen and reported on.

### *Flexible hierarchical Epics, stories and tasks*

Gemini support hierarchical work items, which are commonly referred to as Dependencies. You can create Stories as children of Epics and Tasks as children of Stories. Gemini's flexibility means that it is us to you to define the levels at which to track your Agile workload. By default the Agile Project Template only has Stories and Tasks but you can create whatever Types you need and link them however you see fit. In Gemini you can create a Task as a child of Story A but the parent of Story B, reflecting real world complexities and work orders.

However you structure your Agile workload, Gemini will always roll time and effort up the hierarchy and will enforce the rule that the parent cannot be set to a status of closed unless all of its children have first been closed.

### *Quick Entry*

It is worth pointing out at this juncture that Gemini's [Quick Entry](#) app lets a user enter multiple items in a hierarchical manner in a single data entry screen, applying defaults. Users can then use Gemini's bulk update and inline editing to modify one or more items post creation. This way you can log the work in the standup in its hierarchical order and flesh it out later!

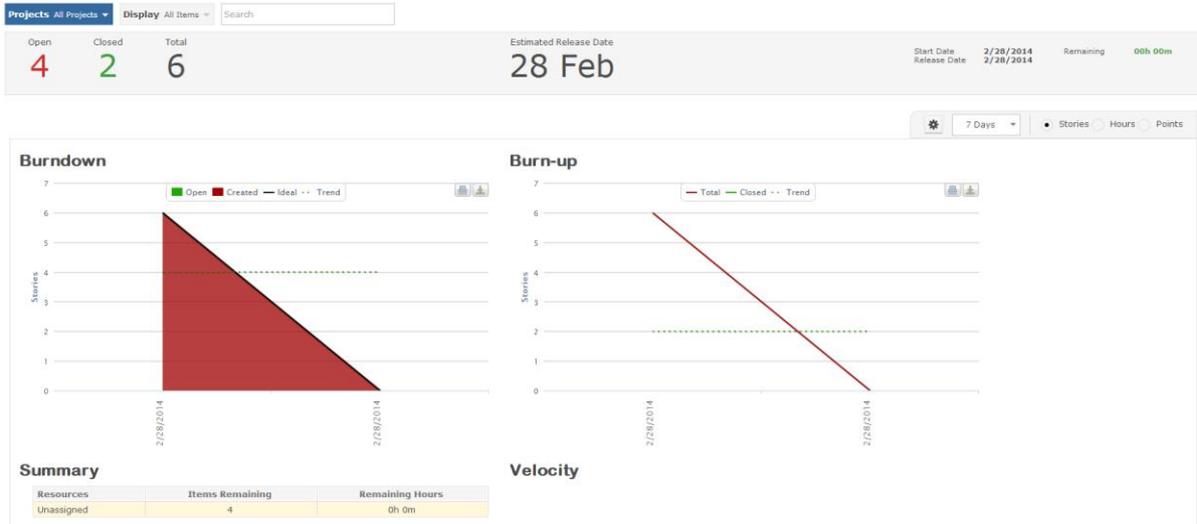


Fig 2.0 The Gemini Progress screen, where Burn-down, Burn-up and Velocity charts are shown

## Flexible Workflow, Rules & Actions

### Workflow

Workflow is configured at the Item Type level; this means that in the same project different Types of work item can have unique (or shared) Workflow. A Bug, for example, can have different Workflow from a Change Request (which perhaps has an Approval status) even though both are tracked in the same project.

Each Project Template defines the full set of statuses that all the different Types of work item in it can pass through. Workflow creation is as simple as defining the list of statuses and clicking on the Workflow hyperlink associated with the Type on the Project Template. Gemini will then 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.

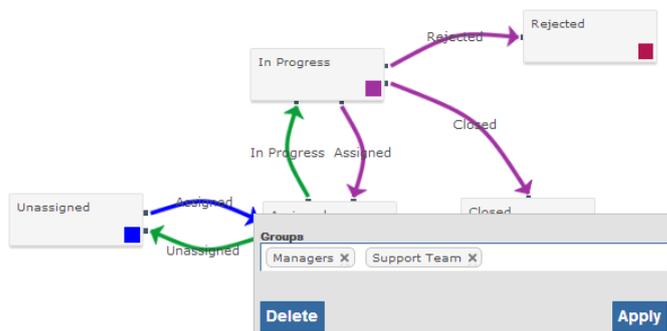


Fig 3.0 A Workflow snapshot showing a status transition that is restricted to specific User Groups.

## Rules & Actions

The ability to implement data-driven logic without programming is a powerful aspect of flexibility because it lets users define the behaviour of the system in a manner that was not pre-determined. Rules and Actions are very simple and you can find a couple of brief video examples [here](#) that show how easy they are to set up.

Rules test a range of conditions, Actions are the things that Gemini can do as a result of matching the condition(s). The tables below list conditions that can be tested and actions that can be taken.

Condition	What Gemini Checks
<b>Comment</b>	The contents of a comment
<b>Commented</b>	The item has a comment created
<b>Comment Source</b>	Who created the comment
<b>Comment Source Type</b>	How the comment was created. E.g. check if comment created by Email Reply (Email) or an email converted to a comment (Breeze)
<b>Description</b>	The description
<b>Due Date</b>	The value of due date. You can specify dates using built-in special keywords with + and - operators e.g. start-week or end-month+10 etc.
<b>Emails Count</b>	How many emails sent to a customer using Breeze Email Reply
<b>Issue</b>	Check if the issue has been created or updated
<b>Priority</b>	The value of priority
<b>Project</b>	If the item belongs to one or more specific projects
<b>Resolution</b>	The value of resolution
<b>Resource</b>	The name of the user who is assigned to the item
<b>Severity</b>	The value of severity
<b>Source</b>	Who created the item
<b>Source Type</b>	How the item was created. E.g. check if the item was created from an email (Breeze)
<b>Start Date</b>	The value of start date. You can specify dates using built-in special keywords with + and - operators e.g. today or start-week-2 etc.
<b>In SLA</b>	Check if the item is in an SLA
<b>SLA</b>	Which SLA an item is part of
<b>SLA Time Left</b>	How much time is left before the SLA is breached
<b>SLA Time Passed</b>	How much time has passed in the SLA
<b>Title</b>	The value of title

<b>Type</b>	The value of type
<b>Name of Custom Field</b>	The value of the custom field name

<b>Actions</b>	<b>What Gemini should do if one or more conditions is met.</b>
<b>Email</b>	Send an email to the email address of your choice using an Alert Template of Generic Type. Gemini will only send one email per item
<b>Set Field</b>	Set the field value
<b>Set Custom Field</b>	Set the custom field value
<b>Veto Action</b>	Stop the creation or update of an item
<b>Stop Clock</b>	Stop the SLA timer
<b>Pause Clock</b>	Pause the SLA timer
<b>Resume Clock</b>	Resume the SLA timer

If the chosen action is to send an email then given that you might not want to hard-code the email address of the recipient Gemini allows you to use tokens that will be dynamically interpreted. The following tokens are supported:

<b>Token</b>	<b>Email Sent To</b>
<b>{resource}</b>	Email the users who are assigned to the item
<b>{reporter}</b>	Email the user who has created the item
<b>{follower}</b>	Email the users who are followers of the item

### ***SLA – a special Rules and Actions combo***

SLA, the starting and stopping of a clock in response to business-specific conditions is a perfect example of Rules and Actions (Gemini SLA is an extension of its Rules and Actions engine). Take the ability to do the following:

*Start a 1 hour response clock if a support ticket is created from the email address boss@company.com but an 8 hour response if it comes from anyoneelse@company.com. The use of an email address to determine SLA response times is not something that could (or should) be pre-determined because you might not have any such business rule in your organization. In your organization you might want to start different clocks based on Priority or Due Date.*

**Note:** You do not have to have SLA as the only reason to start and stop clocks, Gemini will let you deploy this type of logic on any project and any Type just as it will let you create items of any type in any project from emails, not just support tickets.

## Handling stakeholder jargon - Template Taxonomy

For ease and breadth of adoption Gemini can handle the jargon of multiple stakeholder groups in a single instance. Custom fields pose no problem since they can have names that reflect the user jargon but built in fields that are tied to specific functionality (like Resource, Start Date, Due Date etc.) can also be renamed.

Every Project Template is associated with a folder on the web server. This folder can be found at:

App Data\Templates\<<Template Name>

In this folder there is a resource file, within which you can select any of Gemini's fields and override it with your own terminology. For example you might decide to rename Priority as Risk, or Resource as Assignee.

Because you override taxonomy on a Project Template basis you can have different terminology for the same field in the same Gemini instance. For example, users in a marketing department can view and manage their issues and tasks and refer to discreet chunks of time as Week Numbers, or even Campaigns. In the same Gemini instance the software developers can refer to their discreet chunks of time as Sprints. All these terms result from overriding the default Gemini term for a chunk of work executed in a period of time, which is Version. If you have access to the file system of the web server and navigate to \App Data\Templates\Agile and edit the file Resource.xml you will see the override of the default term Version to be Sprint, and the term Item as Story etc.

For more information on managing taxonomy on Project Templates and to see a video of this being done see the documentation [here](#).

### The Workspace override

That every project is based on a default Project Template does not mean that users who have to work with its items are limited to the jargon of that Template. One of Gemini's unique features is that it is Workspace based. Data is held in projects because that is the level at which permissions are set, but users work with Workspaces because that is the level at which functionality is selected, Workspaces can span projects, and Workspaces *let you choose which Template to take the taxonomy from*, which is only natural given that a Workspace could pull data from multiple projects.

### Smart Data Capture

Smart data capture involves having a standard set of fields that cover almost every eventuality, a built-in set of behaviours around key fields, field-level permissions to define who can see what and when, and the ability to create Custom Fields when there is some need to capture data that is unique to your circumstances.

### Standard fields and built-in behaviours

When it comes to bug tracking and issue tracking, standard fields like Priority, Severity, Start Date, Due Date, Estimates, Time Logged etc. are all pre-defined in Gemini. So too is field behaviour around some of these fields. By field behaviour we mean things like:

- Comments that can be marked as visible only to specific User Groups
- A dynamic Source field that drills directly to a failed Test Case if the source was a failed Test or holds the email address of the ticket creator if a Ticket was created from an email

- Computed values like Time Remaining, which is the difference between estimated time and time logged on a work item  
etc.

**Note:** The full list of standard Gemini fields can be seen on any Project Template by clicking on the Screens hyperlink for any Type (e.g. Bug, Ticket etc.). All standard fields are available on all Types.

### Field level permissions

Screens list fields in 3 columns, allowing you to choose the fields that are available when one is creating an item of a given Type, editing an item, or viewing an item. All one has to do to enable a field in one of those modes is tick the checkbox beside it.

Next to each field is a person icon. Clicking on this lets the user select User Groups that can access the selected field; it is also where a field can be specified as being mandatory. If a user is not in a User Group that can access a field then they will not even see the field. In this way Gemini lets you specify who can see and interact with which field in different modes. If a field is marked as mandatory then Gemini will only require a value from those users who have access to the field.

For more information on field level permissions and how they are set, see the documentation on Screens [here](#), which also has a video showing a Screen being set up.

**Note:** Under creating, editing and viewing you can see more than simple fields. For example Dependencies is not a simple field, neither are attachments, but by treating them as if they were, Gemini provides a consistent administrative interface about what can be created, edited and viewed.

### Custom Fields

No matter how comprehensive the out-of-the-box data set there are usually one or two pieces of information that are unique to an organization and than means Custom Fields. Gemini allows you to define Custom Fields of every type – numbers, text, rich text, dates, checkboxes and more.

There are two classes of Custom Fields: simple and complex. Simple Custom Fields are just additional data points – a User Picker (which lets you define the User Groups to pick from) for approvers, or a Number field to hold a budget etc. Complex fields are things like Gemini's Cascading Custom Fields, which allow you to base the values of one selection list from another, for example by making Zip Code cascade from City, when a user selects a City only valid Zip Codes from that City are available to select from.

For more information about Gemini's basic, dynamic and cascading Custom Fields, see the documentation [here](#).

**Note:** In references to Gemini's ability to show lists of data from SQL tables, the term 'table' is synonymous with a SQL view, which can be a view into an external database. You could, for example, create a Custom Field for Customer that presented the user with a dropdown list taken from the database of an external CRM system.

### Alternatives to manual data input

Defining Screens and permissions for users who log in to create and manage work items is simple but there are occasions and scenarios where the data capture starts with an email or a point of integration with a 3<sup>rd</sup> party system (e.g. an Intranet site).

### *Email (POP/IMAP and Exchange)*

Gemini's app framework is not just for 3<sup>rd</sup> party extensibility; one of the key apps is Breeze, which provides email-to-ticket conversion. Breeze will process any POP, IMAP or Exchange mailbox that is configured, executing pattern matched exclusion, inclusion, truncation and attachment rules. Emails that pass its filter will be turned into work items, most commonly but not exclusively support tickets, with values other than Title and Description being populated by Gemini's ability to support Type-based defaulting and the Rules and Actions (incl. SLA) engine.

### *API*

Where there is a need for system-to-system integration Gemini provides a comprehensive RESTful API. One of the world's largest data processing software providers uses Gemini's API to automatically create Tickets when data exceptions arise.

### *Using the apps framework*

That Gemini uses its apps framework to deliver functionality such as Breeze, Sentry (Test Plans and Test Cases), AD integration and more, is an indication of how robust and malleable this framework is. You can build Timer apps, which execute on a user-configurable schedule, Event apps, which fire in response to events (such as an item being created/updated/deleted) and full-scale UI apps, which give you data capture and processing capability, like the Testing screens in Sentry or the Quick Entry screens that let users create hierarchical work items in a fast, simple manner.

## **Security and Scalability**

### **Security**

Banks and financial institutions (including at least one Central Bank) in the Fortune 500 and FTSE100, healthcare providers (including the largest in the world), and national/state institutions rely on Gemini to keep their data safe. There is little more to say about its security other than it has passed every security audit these customers have ever demanded of it and it implements all the security features you would expect - password expiry, password strength, password history rules etc. - as a matter of course.

The more subtle aspect of Gemini security is how it works within the application. We have said that it implements field-level permissions and the ability to control comment visibility, but there are three further security features worth mentioning:

1. Workspaces do not override project or field permissions. For example, a Workspace that spans Projects A, B and C can be shared between users who have access only to Project A, those who have access to B and C and those (the Workspace owner for a start) who can access data in all three projects. Users see what their permissions let them see, and not necessarily what the scope of their Workspaces cover.
2. Gemini has a View Own Items Only permission setting that prevents users in selected groups from seeing anything other than the work items they create or are assigned to, irrespective of any other permission they may have. This means that it is possible to let 3<sup>rd</sup> parties and outside stakeholders into shared projects secure in the knowledge that they cannot see data they don't 'own'.
3. Gemini's data export (and therefore reporting) checks project and field security and does not export data that users do not have the right to see.

## Scalability

Large-scale organizations are not the only users of Gemini, it is a product that is essential to many small and medium-sized teams around the world. Gemini has always been free for up to 3 users a fact that thousands of teams have taken advantage of. Scalability lies in the fact that it is the same codebase that these teams take for free that powers large enterprise deployments with thousands of users.

### *On-demand or On-premise: your choice*

In decisions on scalability it helps that Gemini is one of very few products in its space that is available in both on-demand (hosted by us) and on-premise implementations. Lots of people ask what the difference is; it is not the code, which is identical. Usually the requirement for on-premise is based on the following criteria:

- Corporate policies that preclude all SaaS deployments
- Corporate policies which dictate the use of Active Directory for managing users
  - If you use AD then you must deploy on-premise, it is too onerous to have to punch through all those firewalls and we cannot maintain a customer's AD

***Note:** Gemini support Windows authentication even if you do not use Active Directory; you simply have to create the users manually. If you have AD then you naturally have Windows authentication. The other authentication mode is Forms (username and password). Gemini does not support both Windows and Forms authentication in a single instance but you can have as many instances of Gemini as you like pointing to a single database. Many customers have AD/Windows authentication for internal use and deploy a separate Forms authenticated instance, pointing to the same database, for 3<sup>rd</sup> parties. You need a separate license for each Gemini instance and second/third... licenses are sold at a percentage of the main license, which must have more or equal numbers of users.*

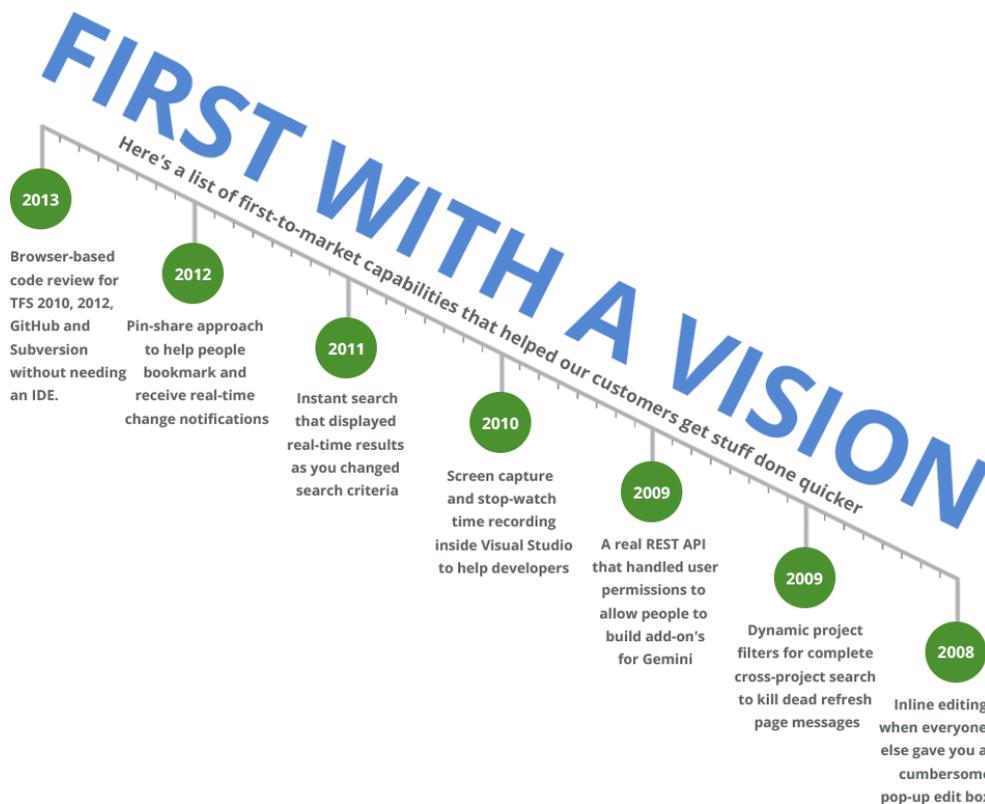
For more on deploying multiple instances of Gemini in a high availability environment read the Gemini High Availability Guide [<hyperlink to pdf>](#).

## Gemini's Technical Architecture

When you read about all the features and power of Gemini it is tempting to think that it must be a huge application. It is, but not in a bloated sense. The easiest way to tell how well architected a software product is, is to check the ratio of its feature set to download time. You can try this on Gemini yourself; just go to <http://www.countersoft.com> where you will find a link to the latest release of Gemini in the footer of the homepage. If it takes more than a minute to get your hands on the most advanced .Net, Enterprise grade Ticketing and Tracking, you need to have a word with your network team!

## One of a kind

When it first came out we released a blog pointing out that Gemini was the first commercially available MVC4 application in the world. Our list of firsts is impressive as the following infographic from our website shows.



Gemini is .Net, runs under IIS framework 4 and works with SQL Server, MySQL and SQL Lite databases, the last of these we deploy only on hosted trial sites for reasons of scalability.

## Get your hands on the code

For those that really need to make Gemini their own there is the RESTful API, apps framework with almost all of the Gemini apps being [open sourced](#). Customers who purchase the Enterprise edition have also earned the right to purchase the Source Code to the core application. For a .Net development team there is no better Issue Tracker or Bug Tracker to get your hands on if you need to integrate with other systems.

## UI, UX and Reporting

### The best UI/UX

The most functional grid of any application in the world, the slickest planning board with a host of features that no other product can match, Workspaces, slide out toolbars with instant metrics, ad hoc and reports, a people view of workload and instant chat with Workspace members... It is quite simply a list of unique features that delight all of our customers and make Gemini one of the best kept secrets in the world of Issue and Bug Tracking, where 90% of sales come from referrals.

How often have you seen a grid that has Zones, dynamic columns, collapsible hierarchical rows, an SLA clock section, inline editing so that you can edit whatever you click on (if you have the permissions), badge count notifications, chat message notifications etc. That is just the Items Grid. The Board has an equally impressive list of UI and UX features. Independent swimlane scrolling so you can manage large backlogs, dynamic columns that respond to whatever you axes contain, dual axes so you can slice and dice, four levels of zoom... There is more but already Gemini has left the rest trailing in its wake.

### Simple, powerful ad hoc and standard reporting

All Gemini's reporting is done through Excel because it is quite difficult to imagine anything that can be reported in a better format. Whether you want summary reports, with the ability to drill to detail, charts, or just lists, Excel is the ubiquitous standard. There are very few teams that will struggle to find someone who can build a Gemini report.

Gemini's will export to Excel (either a pre-defined report or a data dump) whatever is filtered by the Items Grid, whether it is in view or not. This export respects permissions of the user who is generating it so security is never bypassed by reporting. Finally, Gemini Workspaces, which dynamically filter data, can be set to produce any reports on a schedule and to distribute them to individuals or groups of users.

For more information on Gemini's Excel reporting, including how to create your own reports, see the documentation [here](#).

## A complete package

Seeing, touching, believing. At Countersoft we have a mission to run counter to the standard software industry's *money for old rope* mentality. Look at the rate of new releases from our News section to see how hard we constantly strive to produce better quality, innovative solutions for our customers. A complete package starts with a vision, ours is to give you the best tracking and ticketing solution in the world, so you can get the job done right and go home on time.

[Videos](#)

[Documentation](#)

[Support](#)

[Download our EULA](#)

[Contact us](#)