PAINLESS TIME MANAGEMENT

COUNTERSOFT

Contents

Painless Time Management

What we're going to cover

Managing time is not one of those exciting aspects of our working lives. It can be painful to find that the most persistent quality of time is that there isn't enough of it. Gemini takes the pain out of time management, giving you back hours of your day. Collated over the potential number of individuals involved, Gemini delivers significant value-for-money.

In this 'guide' we're not going to simply state that Gemini lets users log time against tasks, instead we will follow a typical problem from its moment of recognition right through to closure. In the process you will see how the versatile time management features of Gemini pro-actively manage time for you, so you can go home on time.

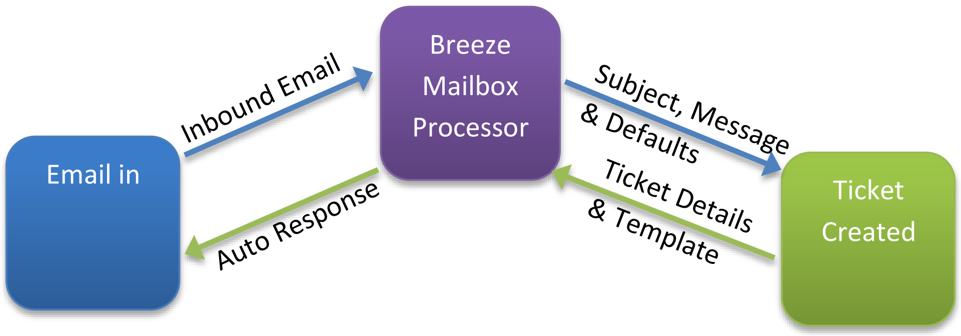
Customer finds a problem

Jody, a customer of Fenrise LLC, finds that for some inexplicable reason their software is not performing a complex calculation properly.

Jody emails support@Fenrise.com to let them know a) she found what she thinks is a bug and b) she needs a fix pronto.

Preventing noise

Fenrise use Breeze, Gemini's email-to-ticket app that comes out of the box. Their Breeze configuration has the all-important Auto Response enabled, which is connected to an email template that lets Jody know that her email has been received, it is being dealt with, and a technician has been assigned to the case. This last part is only partly complete, this company triages inbound support requests and a CSR will examine Jody's problem and assign a technician.



Jody has the satisfaction of knowing that her problem is in hand and she has a ticket number so that all future communication around this issue can be tracked. For their part, the support team have an instant point of reference if she calls or emails again. Jody has no reason to do what people do if their communication appears to have gone down a black hole, phone up and escalate her problem as high into the management hierarchy as she can reach.

Time management starts

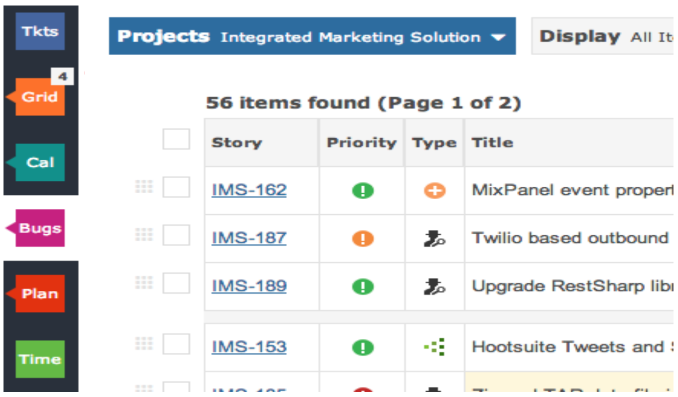
Fenrise has enabled Rules and Actions with a rule that recognises Jody's inbound email address. The domain of Jody's company is a value in a Custom field that the rules check for email conditions. The rule matches the domain against the inbound email address, sets the Priority of the ticket to High and starts the VIP Customer SLA, setting the response time to one hour before the ticket is in breach.

Gemini timestamps the ticket on creation and tracks its source. From the very beginning, time and communication channels are monitored. Incidentally, Gemini is an international product where dates are stored in UTC format, so it does not matter where in the world anybody in the process is, time on the Ticket is tracked universally.

Workspace Magic

Jim works in Customer Support. One of the Workspaces in his Workspace collection is the High Priority Workspace. This particular Workspace filters items from all support projects that are in a Status of Open with a Priority value of High.

When Breeze creates Jody's Ticket, Jim's Workspace card lights up with a badge count notification to tell him that there is work for his team to deal with. If he has set up Alerts for when items are Created in the scope of his Workspace then Gemini will also send him details of this item by email and if he has subscribed other people or groups to alerts from this Workspace then they will be notified too.



Projects Integrated Marketing Solution | Display All It

56 items found (Page 1 of 2)

Story	Priority	Type	Title
IMS-162	●	●	MixPanel event propert
IMS-187	●		Twilio based outbound
IMS-189	●		Upgrade RestSharp lib
IMS-153	●		Hootsuite Tweets and

fig 1.0 Workspace badge notifications tell users when items they have not seen come into scope

Auto Notifications

Jim looks at the Ticket and decides it needs to be handled by an expert in the Accounting Systems division. Using Inline Editing (point-click-change), he edits its status from Open to Assigned and updates the Resource field to assign the Ticket to Jack.

Fenrise LLC have configured Rules and Actions so that the change of status from Open to Assigned pauses the SLA clock. They have another Rule and Action combo that restarts the clock, emailing the resource and a Manager if the ticket is not dealt with in a timely manner. *Whatever the business rules are surrounding SLA escalation and management, Gemini should be able to deal with them under Rules and Actions.*

Auto Alert for Resource Assignment is enabled, so Jack receives an email from Gemini informing him of his assignment. It is a redundant message, he has his own Workspace, which filters work using the meta filter "Assigned to me" so his badge count notification lit up and the Ticket appeared in his list of tasks in almost the same second that Jim pressed Save. Gemini understands the token "me" to mean the Workspace user, so if this filter is set up and the Workspace shared every recipient user of the Workspace will see items assigned to *them*. In the same way Gemini understands other tokens, such as “TODAY”, “START-MONTH” etc. as dynamic values to be interpreted at filter execution time.

Jody too will be notified of the change of status because Fenrise LLC has enabled the Auto Follower Event app, which marks the Item Creator and any Resources as Followers as soon as an item is created.

Gemini is keeping all stakeholders informed and not relying on human-to-human communication. Clocks are being controlled automatically by business rules that respond to conditions.

Continuous Communication

Jack looks at Jody’s description of the problem and decides that he doesn’t have enough information to solve it for her. Using Breeze Reply he sends an email back to Jody requesting more information. He doesn’t need to know her email address, Gemini tracks it automatically in the Source field and will also track anyone who is CC’d.

Note*: You can configure Breeze to have multiple Reply From addresses so that CSRs can respond on behalf of other individuals. You can also use the Breeze Outlook connector so that emails sent to individuals can be converted into tickets from Outlook.*

Fenrise LLC have configured Rules and Actions on their Platinum SLA clock such that the first manual email back to Jody stops the clock. It might not be possible to fix Jody’s problem before Year End but she has her response and the SLA has not been breached.

When Jack replied, Gemini put the Ticket ID in the subject line and turned his reply into a comment on the Ticket. Gemini tracks email and manual comments separately, even letting you create Rules and Actions around these two different comment types. Jody receives Jack’s query and replies back and the system is smart enough to identify the Ticket ID in the subject line and append all on-going email communication as comments on the same item. This way, anyone looking at the Ticket can see the full history. The “Hide” link above the comments lets the operator collapse all but the comments from view, sorted in descending date order, so it is really easy to get a handle on the latest status of an item by reading the newest comments.

Gemini is keeping all communication together so there is one version of the truth for all parties. Anybody coming into the picture can be brought up to speed at a stroke. If Jody or Jim goes on holiday nobody picking up the problem in their place will be blind-sided.

Transforming Issue Types

Jody’s answer confirms to Jack that she has found a bug. He edits the Type field on the Ticket and changes it to a Bug. Gemini immediately alters the screen to show the fields for a Bug, which are different from those of Type Ticket. For one thing Fenrise LLC has configured their Project Template so that Bugs require a Sprint or Version in which it is proposed they are fixed. Bugs also pass through different statuses from tickets and one of the Rules and Actions kicks in to set the status of the Bug to Backlog when it is Saved. Bugs don’t have Followers. This does not mean that the field is somehow deleted or blanked out, only that the value(s) in the Follower field cannot be seen and Follower behaviour (sending emails to the users in the field) is not implemented. There is such a thing as over communication with 3rd parties.

Fenrise items of Type Bug have two Custom Fields: System Area, a dropdown list, and Customer Account Number, a textbox. Bugs also use the Standard Severity field, renamed by overriding the taxonomy to the word "Difficulty", which Jack sets to a moderate level that equates to a day's work.

Jack updates System Area to General Ledger but he doesn't know the Customer Account Number. Luckily Jody can also log into Gemini and exists as a user in it, and Breeze has been configured to match her email address to her username. All users are authenticated via Active Directory and the AD configuration has been set to pull in the Phone Number, Location, and Customer Account Number of all users. All Jack has to do is to click the plus sign next to the *ReportedBy* field and he can get a pop up with those details.

***Note**: ReportedBy is different from Creator to allow users to create items on behalf of other users. Where the user creates their own items ReportedBy and Creator will be the same.*

Time Log Entry

Jack clicks on the Time tab on the Item View screen and adds the time he has spent analysing Jody's problem against the TimeType Investigation, which is one of the TimeTypes that Software company can subsequently analyse to help tune their business processes. At the end of the week Jack, like all delivery personnel, runs the standard Timesheet report, which exports his time to Excel and produces formatted pivot reports.

Project Managing Issues and Bugs

Bugs and Issues with no associated Sprint/Version light up John the PM's Workspaces. This is work that needs formal estimates and scheduling. Because Jack transformed the Ticket into a Bug all of the information and history is retained. It also took only a fraction of a second to 'create' a Bug in the system because nobody has to input another transaction (possibly in a separate system) for the Software Development and Testing teams.

The default view of John's Workspace is the Board, where he plans all the Sprints for the development team. His Board is set to with Status on the primary axis and the cards are color-coded by Priority. Seeing that he has an Open, High Priority work item he drills into it.

John's permissions let him see everything, including Jack's Difficulty rating. He wonders if Jack's rating assumes that he will (Jack) will be assigned to fix the bug. John slides out the right hand toolbar, clicks on the People tab and sees that Jack does not come free for several days. He clicks on the Chat tab and asks Jack if the rating would be the same if another resource tackled the problem. Jack shares the PM Workspace and he will see the chat notification light up.

Gemini ensures time is not lost in simple, internal communication and all communication is tracked.

Switching tracks

Now the Ticket is a bug the focus of time management is with the software development team. Time is relevant in many ways and its tracking must answer the questions:

- How long will the bug take to fix?
- When is the user going to get a solution (which sprint/version/release)?
- Whose time is going to be invested in the problem?
    - By resource type (PM, Developer, Tester, DevOps)

    - Is that billable (what type of time)
- After the problem is solved can the team's ability to estimate be refined in a retrospective analysis?

A good Gemini configuration just asks for and provides the answers to all these questions

Jack confirms the difficulty rating in his Chat reply, so John drags the card representing the Bug from Status Backlog to Status Sprint. He right clicks on the card and edits the Sprint field to the current sprint, converts the difficulty rating into Estimated effort (hours), adds Julie, a software developer, as the Resource and sets the Due Date to the middle of the week. He does not need to set the Start Date, the Helpdesk project of Fenrise LLC uses Gemini's date tokens for defaults. Start Date is set to default to TODAY, a token value that Gemini understands (among other date tokens) as the date the item is created.

All John's changes are done using inline editing - click on the fields and change the values, from lists if they are lookup fields.

Before he finishes with the task, John enters his time log for the brief amount of Time he has spent on it.

Fenrise LLC use Agile as the methodology in software development and have deployed the Gemini Roadmap and Changelog apps. Adding a new item to the current sprint will automatically be highlighted on the Burndown and Burnup charts of the Roadmap, since that is the kind of thing that will potentially disrupt the delivery schedule. Fortunately Gemini also plots Velocity on its Progress screen so PMs can see at a glance if they have historically been able to cope with the workload that the current sprint/release contains.

Gemini is now tracking the Created date, Revised date, Estimated Effort, Time Logged, Time Remaining, Start and Due Dates and likelihood of meeting a delivery schedule – painlessly.

Keeping the Customer in the Loop

So far everything that has happened since Jody's email created a Ticket could have taken minutes but as the user affected she should be kept appraised as soon as information relevant to her is known or changed. She is no longer a Follower since Fenrise LLC decided that the field did not apply to Bugs, however if the helpdesk has a Workspace with a filter:

- Project = All
- Type = Bug/Ticket/Issue/etc.
- Status = Sprint/In Progress (statuses are drawn from all projects, not just software dev)
- Due Date <> Empty
- Customer Account Number <> Empty

…then it will tell them whenever work for a Customer is ready to be dealt with by a delivery team and a Due Date has been assigned.

Jim sees the Bug in his customer notification Workspace and emails Jody to tell her that an issue has been found with the product, the team are working on a fix and it is scheduled for release to her on the Due Date.

Note: *This notification could be done automatically using the Workspace Alerts and the Workspace itself. If Jody was in a User Group that had the permission View Own Items Only and the Workspace was shared with the System Group Everyone then Jody would share this Workspace but would only see the Ticket she raised, now transformed into a Bug. Of course it would be easy to hide fields that she shouldn't see in that transformation, because field level permissions on item Type Bug could state that only CSRs, Developers and PMs could see* ***Type, Resource, Due Date*** *etc. …and the team could all communicate with comments that were marked for internal groups only, keeping users in the chain only when they want to.*

3rd Party Integrations

Fenrise are a .Net outfit and their developers use TFS and Microsoft Visual Studio, with which Gemini has out-of-the-box integrations. They could be using Git, Mercurial, SVN or other ALM tools, Gemini has lots of integrations. The point here is that the Julie does not need to log into Gemini to see her work assignment, she filters tasks assigned to her directly into her Visual Studio IDE. She also does not have to calculate how much time she spends over the next few days working on her task, she simply starts a clock when she starts working, pauses it when she takes a break, restarts, and stops it hen she's done. At that point she can allocate the time to a task or multiple tasks. Gemini will even pro-rate time for her if she just needs an even split across multiple tasks.

Tracking Testing Time

When Julie has completed and unit tested her work she can click on the item in Visual Studio, drill into Gemini and update its status. She does so, changing the status to Ready for Testing.

This new status lights up a Workspace badge notification for the Testing team and the item is now pulled into their Workspace.

Note: *As items change their properties the filters that draw them into Workspaces may no longer apply. For example, the filter that first pulled Jody's Ticket into the CSR view may only have worked on Tickets, or Tickets in given statuses. As soon as the item was turned into a Bug it would automatically leave that Workspace as it is out of scope. The work item is not hanging around in all the Workspaces it transitions through (unless that is desired in the way the filter is set up) it is moving from group to group*.

The Testing Team review work that comes into their Workspace and decide whether they should be tested in situ or moved. Items can not only be changed in Type, they can be moved to different projects *and* become different Types. The testing team of Fenrise LLC have a TEST project, based on the Sentry Project Template within which one of the Types a Type Defect.

Jody's problem is deemed to be a regression test issue and so it is moved to the tester's project and turned into a defect, with all the accompanying changes of fields and permissions that this entails. This too could light up notifications and send emails to CSRs and PMs.

The fix is tested using the Sentry app, which like Breeze is delivered out-of-the-box in every Gemini installation. Sentry has Test Plans and Test cases so that when regression testing is performed nobody needs to reinvent the wheel. The new Defect is attached to a Test Case and tested. In the process if any issues arise the tester can:

- Mark the test as failed but proceed to the next Test Case in the Test Plan
- Mark the test as failed and fail the run (a full history of every test run is kept)
- Mark the test as failed *and* raise a new issue of any Type in any Project

Fortunately the test is successful, the item is changed back to a Bug and moved back to the Software Dev project with a Status of Ready to Deploy.

The software tester enters a Time Log for the time taken to test the defect.

Closing out Bugs and Issues

Ready to Deploy is quite naturally the status selector for the DevOps team's default Workspace. They now have to schedule a release of the entire software package, together with release notes about what has changed etc. They will mark all items in the release as closed simultaneously using the Items Grid filter to select all items in the Sprint and then use its Bulk Update capability to change the status on all of them to Deployed. Deployed is a Status that Fenrise LLC has defined as a Final Status and therefore it is synonymous with closed.

Using the Timer app framework and generic Email Templates, Fenrise LLC has written its own post deployment integration, something that is simple to do especially since they could base the code on the Open Source apps which are all on GitHub. Their app scans all items in a status of Deployed, groups them together by Customer Account number and sends an automatic email notification to their key accounts letting them know which of the issues they reported have been fixed and where they can download the new release from. No time wasted there.

Sprint Retrospective

Simply because an issue has been solved does not mean the end of the process. Agile software development teams will reflect at the end of each Sprint Cycle on what they did, how they could improve and what lessons have been learned.

All the time that the Sprint was open it could be tracked on the Progress screen with Burn-down, Burn-up and Velocity charts. Here the PM could track Estimates against Time Logged. On the Items Grid and Items views it is easy to amend percentage complete and to see the hierarchical breakdown of items with dependencies. If Jody's bug had been related to other items in the Sprint it could have been made a parent or child of another item, irrespective of that item's type. Of course Gemini consolidates Time Logged, Estimated Time and Points (if you use Points for estimating) up the dependency hierarchy, so that you can see these values for each item and at an overall level.

Open items in the Sprint can be easily seen along with Summary data by deploying the Roadmap app, a simple way to see scheduled work that is current and future by Sprint/Version/Release. Once the items in the Sprint are marked as closed, irrespective of the name of the Status, they move to the scope of another app that is frequently implemented in Agile teams, the Changelog.

Gemini tracks time not just as simple time entries on timesheets but as time initiated, time to respond, time to execute, estimated effort, actual effort, and after the task is complete you can review the Changelog, examine the time spent that it encapsulates and use the learnings to improve your products/services and increase efficiency.

Seeding the Service Desk

If you have an ITIL/ITSM initiative and/or a Service Desk in which you capture strategic initiatives for business process improvement, you can simply implement the Gemini ITIL Template and transform resolved problems into an open initiative, or create a new one and use the Item Linking feature to base it on the original problem with its rich history. For more information on connecting Helpdesks and Service Desks see The Ticketing and Service Desk Field Guide <Hyperlink this document>.