

## Wishlist Team

The info below should be moved to [Wishlist Team](#) as per [Where](#) (because this page is not feature requests and bug reports about Tiki the software, but some ideas on how to better use Tiki for the community).

These are shared procedures/instructions for all the people monitoring Tiki wishlist



### Real time Tiki ticket statistics

## Who

See: <https://tiki.org/Wishlist-Triage-Team#Who>

## What do coordinators do?

- [Click here to monitor all changes to the wishlist](#) You will get quite a few emails. You can turn off later.
- Check all new submissions. (people use [Make a wish](#) and read [How to Submit a new item on the Wishlist](#) so make sure these pages are always clear.
  - Add any missing categories
  - Assign to people who have expressed that they want to maintain feature X or Y or who have caused the bug 😊
- If there is a [security](#) issue reported, make sure it doesn't disclose the vulnerability until a patch is issued.
- Invite people to contribute directly to Tiki source code, especially if people submit a patch
- prepare next version

Below are some some explanations for usage of fields not on [How to Submit a new item on the Wishlist](#)

## Status

Open: Needs some love “ No decision made yet

Pending: Fix is on the way but not yet in a stable release OR there is another issue.

Closed: Solved and available in latest stable release OR will not be fixed / is rejected / etc.

## Rating

Add points (+1 or +2) to bugs which have affected you and to features which you would find useful. You can also vote -1 or -2 to bugs which bugs you can't duplicate and to features which would be a waste of developer time. This will help us all.

## Resolution status

- Accepted
- Confirmed
- Duplicate
- Fix on the Way
- Fixed
- Invalid
- Later
- Not enough information

- Out of Date
- Please retest
- Postponed
- Rejected
- Remind
- Won't Fix
- Works For Me

Assigned to

Don't assign to other people unless you think they agree.

Areas

Only check one if it feels logical. This is not really used at the moment.

Related project

This is used to regroup tracker items into larger project. Think "big picture"

- [AdminUIRevamp](#)
- [Edit User Interface Revamp](#)
- [Infrastructure Revamp](#)
- [TwoRevamp](#)

Ideas to improve the process

Ticket rewamp phases

### **First phase - update the tracker structure**

- introduce ticket types as described below
- make group assignment available (in line with community group structure)
- rename resolution status to "status"
- rename report type to "category"

### **Second phase**

- rework prioritization principles
- introduce queues as described below
- Review and update tickets > all ticket should be put into a ticket type, in a queue, assigned to a group and have a status
- revise categories > create at lease top level and one sublevel
- refine groups

### **Third phase**

- we'll see..

Work in progress, add your ideas!

## About ticketing in general

The main idea mostly is to have a first line of support (guys there are not really experts, but can make some basic decision and channel the tickets to the right place), they handle the first contact, tries to resolve as much as possible to let the 2nd (more experienced supporters) and the 3rd line (programmers, area specialists, etc) work on only what they are really good at. This structure usually enables to respond to and resolve any request in a timely manner.

Of course Tiki is not a company and the community is not a servicedesk, so it should be a bit different, no complicated SLA or other things, but maybe we can incorporate some of those practical and proven ways and maybe follow some guidelines of [ITIL](#).

Tiki should attract more "1st line supporters" who would like to do something but don't know the code and the implication on coding hours on the requests. 1st line could be a group where folks can volunteer (subscribe to group). The rest of the lines is not so open in order to avoid ticketing becoming a mess.

## About ticket ownership

Tickets are always assigned to a group than the current group leader (can be rotating every week/month/etc) would assign the ticket to a member of that group. A ticket should always be assigned to a person in the end. If the member does not want to deal with the issue, he can put it back to the group pool or assign to another group.

## About ticket types

Instead of what we have now (*Report type*) we should introduce basic ticket types, such as:

- incident** for something that is not working as expected. open for anonymous/registered users?. (it can be a bug or incorrect usage of the feature). Can turn into a simple usage question, or a bug/problem/request
- request** (sometimes also called as IMACs) things are working fine but I want to Install/Move/Add/Change something. Open for registered users
- problem** for recurring incidents are collected under a problem ticket. Wishlist squad members can open. It can be fixed or closed as a known error (something we know of but we won't fix, for example at older releases)

Having these basic types makes reporting and channelling easier.

## About ticket categorization

After the ticket type selection should come the categorization itself. (bug subtypes, etc). Here should we have what is now "report type"

The problem here is that there is no "dependency" possibility in trackers (yet). I mean that if "Incident" is ticked that only those category options should appear that are dependent on the "Incident" checkmark to be ticked. I am thinking about the way the main administration option as working now in tiki - only when you put a checkmark into the box the related further options appear.

Category system is usually 2-4 levels deep. Tiki is quite complicated, so the depth should be discussed.

An example:

<b>Field name</b>	<b>Description</b>	<b>Field type</b>	<b>Input type</b>	<b>Example</b>
Category	The top level	drop-down	manual selection	<b>Tiki Feature</b>
Subcategory	The level below category	drop-down	manual selection	<b>Forums</b>
Area	The level below subcategory	drop-down	manual selection	<b>Forum post</b>
Detail	The level below Area	drop-down	manual selection	<b>Posting error</b>

The above mentioned dependency logic should be applied for the category selection too (if I select a category only those subcategories should be available that are defined as subcategories of that given category)

About queues, groups and users

**Queues** are the place where you keep the tickets while they are being worked on

**Groups** monitor the Queues (usually a queue has a default group that is primarily responsible for the tickets in the group).

**Users** are members of the groups and have access to the queue through group membership

Concept of queues is sometimes hard to grasp, but it is handy, you can also define permissions for queues, manage better the workload, etc.

### Proposed queues

<b>Name</b>	<b>Description</b>
First line	this is where tickets arrive upon first save. they stay here until being evaluated by the wishlist squad
Bugfix	this is where those tickets should go that are addressing a bug/malfunctionality and stay here while being worked on
Development	this is where tickets should land when they are accepted as a development
Information	for tickets that are about giving information/training/help
Internal	internal community tickets (assigning jobs)

### Proposed groups


To avoid duplication see here: <http://tiki.org/Teams>

Wishlist should use the same!

About Tiki Release indication

There are checkboxes to indicate the affected Tiki release, which is fine

## Ticket statuses & status flow

Tickets **usually** have the following path\*.

- ticket opened by customer or servicedesk 1st line staff
- servicedesk 1st line tries to make a first contact resolution and close the ticket

First contact resolution percentage is an important metrics of an effective helpdesk. If it is high it means that the info is distributed well, easy to find and understand for non-technical guys

- if 1st line can't resolve, first it tries to collect as much info (symptoms/description) as possible to help the work of 2nd line. For certain type of tickets/categorization you define the necessary minimum information (mandatory fields), they can have certain templates assigned. Where to dispatch than is decided by the rules associated to the type&category combo of the ticket.
- Dispatch goes to a queue (which has a default group), not to person
- if 2nd/3rd line finds the categorization right than first it accepts (it is an event), than one group member "takes" it and starts working. When he thinks the ticket is resolved it goes back to the 1st line who is taking care of the confirmation/closure
- if 2nd/3rd line dont find the dispatch (assignment) correct, send it back to 1st line or if assigns to the correct group

Status name	Old status name	Description	Can arrive from	Can go to
New	-	This is the status upon first save	-	Open - Accepted Pending - Information %% Pending - User test
<b>Open</b>				
Open - Accepted	Accepted	The wishlist team reviewed and the ticket seems to address a valid issue. Mainly to rule out invali	Open - New, Pending - Information	Open - confirmed
Open - Confirmed	Confirmed	The issue is confirmed		
Open - Work in Progress	Fix on the way	The ticket is being worked on by the responsible team		
<b>Pending</b>				
Pending - Information	Not enough information	Not enough information provided to continue with the resolution		
Pending - Postponed	Later, Postponed	issues that are accepted but no resource to work on		
Pending - User test	Please retest	The ticket seems to be resolved, the users should test.	Resolved statuses	Closed or back to the dispatch group
Pending - Review	Remind	The ticket is pending too long, it should be reviewed		
<b>Resolution</b>				

Resolved	Fixed	The responsible team believes the issue is answered/fixed, needs confirmation from the ticket opener before closed		
Resolved - Refused	-	The issue is refused, it will not be done for whatever reason (it is against the tiki policy, no time, no volunteer)		Closed - Refused
Resolved - Works fine	Works For Me	The issue could not be reproduced, it seems to be ok		
Resolved - Duplicate	-			Closed - Duplicate
Resolved - Known error	Won't Fix	The issue is a known error, which means that it is acknowledged but will not be fixed due to lack of time, volunteers, etc		Closed - Known error
Resolved - Refused	-	The issue is refused, it will not be done for whatever reason (it is against the tiki policy, no time, no volunteer)		Closed - Refused
<b>Closure</b>				
Closed	-	The issue is answered/fixed and it is accepted	Resolved	-
Closed - Works fine	-	See above	Resolved - Works fine	-
Closed - Out of Date	Out of date	for really old tickets	Resolved - Out of Date	-
Closed - Duplicate	Duplicate	The ticket is a duplicate, must reference the the other ticket	Resolved - duplicate	-
Closed - Known Error	-	See at resolved, we wont fix this and that is for sure	Resolved - Known error	-
Closed - Won't do	Invalid, Rejected	The ticket was found to be invalid	any	-

Wishlist squad reviews new tickets and correct the incident/request setting when necessary

### Notes about ticket resolution & closure

**Resolution** when the person responsible for resolution thinks that he/she resolved the issue. This always needs a confirmation from the person who opened the ticket

**Closure** the person who opened the ticket confirms the resolution.

Usually the idea is to always send out a resolution email notifying user of the that we think his/her issue is resolved and also stating unless he/gets back to us within i.e 5 working days we consider the resolution to be accepted and close the ticket with the appropriate closure code.

**Closed tickets can never be reopened, they are archived.** Maybe a new ticket is opened later with the same issue, the 2 tickets should be linked together as can be identified as a problem (=recurring incident)

#### About trackers

- it is slow
- maybe closed tickets could be archived&moved into another tracker with the same structure, maybe it would speed up the whole thing
- the fields would need a bit of restructuring, will attach a sample soon
- maybe rename "trackers" to "forms", I think it is a better name for marketing too, more easy to understand for newcomers what it is all about. ("Tiki has a built in forms builder, structure your information as want, etc")

#### About priorities

[Ease Importance Priority](#)

#### Mid term goal

I think following some of these points could make Tiki also to a good support system, since it has all you need to make good support, integrated wiki & forum & ticketing, you don't often have that (and this could be offered as a profile too)

ML: Do you have list of feature requests / bug fixes to make this happen? (vs just reconfiguring).

#### Other ideas

**pascalstjean:** After reviewing the current bug reporting tool; I have been interested in creating a step by step bug submitting tool that will allow us to identify who is actually submitting the bug (Developer, User, Manager etc...) Being able to put the person who is reporting the bug in context is extremely important, we as a group will be able to better organize these bugs and hopefully take action.

I hope to be able to work on this tool over the next few months by refactoring the current categories and options found in the Bug Report. I was really intrigued to create this tool using the same technique that CGCom has done using Pretty Trackers for some of their customers.

Once I have a working prototype I will present it to the community for comment and hopefully get some good feedback. If anyone has opinions or other types of idea's that you have please feel free to get in touch with me. I'd love to hear your comments and concerns about the current Bug submitting process.

**SEWilco:** The categorization section above might be able to use [Dynamic items list](#).

#### Alias

- [WishList squad](#)