Plorma Documentation

Release 0.3

Torsten Irländer

November 05, 2015

Introduction

1.1 What is Plorma

Plorma is a webapplication to plan, organise and manage tasks in the software development. Plorma is influenced by agile development methods like Scrum. Tasks can be organised on Kanban boards and the progress of a sprint can be visualized in a Burndown chart.

Although Plorma is especially suited to support an agile software development it tries to be as general as possible to be used in other fields of activity.

1.2 Licence

Plorma is licensed under the GPL version 2 or later

1.3 Installation

1.3.1 Pip

Plorma is available from Pypi:

```
pip install plorma
plorma-admin app init myplorma
cd myplorma
# Adapt database connection etc in the created production.ini file.
plorma-admin db init --config production.ini
# Optionally install the demodata
plorma-admin fixtures load --path /path/to/plorma/fixtures/demo --config production.ini
plorma-admin db fixsequence --config production.ini
# Finally start the server
pserve --reload production.ini
```

1.3.2 Docker

A Docker image including demo data can be started with:

```
docker run -it -p 6543:6543 -d toirl/docker-plorma
```

1.3.3 From source

Please read the REAME coming with the source of Plorma.

1.4 Get the source

Plorma is hosted on Github:

```
git clone https://github.com/toirl/plorma.git
```

General

2.1 Roles

Plorma has two different user roles to distinguish between the tasks of the users of Plorma.

2.1.1 Developer

Developers are the one who work on the tasks. The can edit delete and create new tasks.

2.1.2 Productowner

The role Productowner becomes important as soon as you want to use Plorma in your SCRUM process. Only Users with the role Productowner are allowed to create an edit Sprints.

Dashboard

The dashboard will give you a fast overview over your

1. Open tasks (See *Workbench*)

2. Running sprints (See *Sprints*)

3.1 Workbench

Workbench **4**

Task	Priority
Data submission fails when user is not registered	25
Add missing "name" field into the export form	24
Call Susan and clarify user story for data import	17
Add new ACME export format	None

The workbench is a **listing of the top 5 open tasks** which are assigned to you. The tasks are ordered by the *Weight of the task* (See *Weight of a task*). In case you do have more open tasks they are not listed here to keep the focus on the most important tasks.
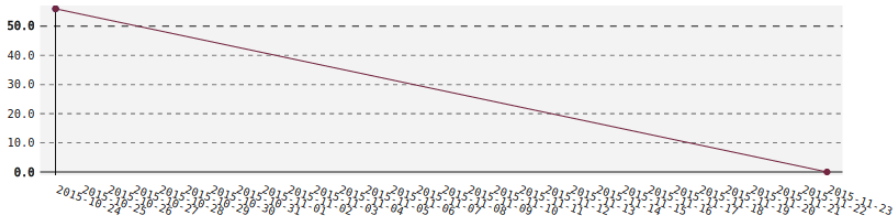
Tip: In sake of transparency it is a good idea to only commit on tasks which you actually can handle in the next time. Otherwise there is the risk that this tasks will never get adressed by other people because you aready grab and "block" the tasks. In order to not get lost in your tasks you should try to keep your assigned open tasks as small as possible!

If you have more than 5 open tasks assigned than the label indicating the amount of tasks will turn orange to indicate a warning. If you have more than 10 tasks open it will turn red to indicate danger to get lost in your many open tasks.

3.2 Sprints

Sprints

Autumn shipment (24.10.15 - 23.11.15)

Strength	100
Initial Story Points	56
Remaining Story Points	56
Velocity	0

⊞ Open Sprintboard

The sprints section will list all currently *active* sprints showing the Burndown diagram and some statistics for each sprint. Active sprint are sprints which are currently in the *running* state.

The *Sprint Board* is reachable by clicking on *Open Sprintboard*. It gives you a more detailed you on the current state of the sprint.

Task

| Plorma | Home | Tasks | Sprints | | 👤 Peter Product Owner (peter) ▾ |

Task: Data submission fails when user is not registered 🔒

| 🗏 | ➕ | 👁 | 🗑 | Advanced ▾ |

Outline
Task
Sprint Planning

Task

Name ✹

Data submission fails when user is not registered

Tags

Defect

Assignee [peter]

peter ▾

State

Resolved ▾

Resolution ✹

works for me ▾

New entry

Comments (2)

Just tested this issue here and can not reproduce it.
15.10.24 08:52 | Peter Product Owner

When user is not regestered the datasubmission in forms does not work correct. Some data is missing in the submitted data and the new dataset can not be assigned correct.

Tested with FF38.
15.10.24 08:51 | Dave Developer

Submit

Name Every task must have a short name which ideally sums up the task in a few words. The name is used in the task overview to identify the task.

Tags The tasks can be tagged with different tags to help organising the tasks. The tags will be displayed in the task overview. Tags can only be created by User with the role *productowners*.

Assignee A task can be assigned to other users. Assigning a task to a user means that this user is responsible to work on the task.

If a task has no assignee than it can be be taken by others.

Task State The *States* and *Resolutions* can be set in . Some of the fields will become a required field based on the current state and selections. So the *Resolution* field will become a required field when switching into the resolved state.

Comments Users can add comments to give further information, or document their proceed. Comments are readable by all users.

Priority The priority (*Weight of a task*) will influence the order of the task in the task overview or product backlog.

Priority	Description
imme-diate	Must be fixed immediately (means: "Drop any other work"). Reports must have an assignee set in the "Assigned to" field.
very high	Should be fixed as next task by maintainers and certainly before the next release.
high	Not the next task, but should be fixed soon. Depending on teams & manpower this can take between one and six months.
normal	Medium priority; would be good to get fixed somewhere in the future. Contributed patches might speed fixing up.
low	This can be fixed, but we're not going to worry about it. Patches very welcome and required for progress.
very low	This can be fixed, but we're not going to worry about it. Patches very welcome and required for progress.

Usually only the priorities from *very low* to *very high* should be used for task planning.

The *immediate* priority is a special one. It should only be used in very rare cases as it really means that any other work should be dropped which may affect running sprints.

Severity

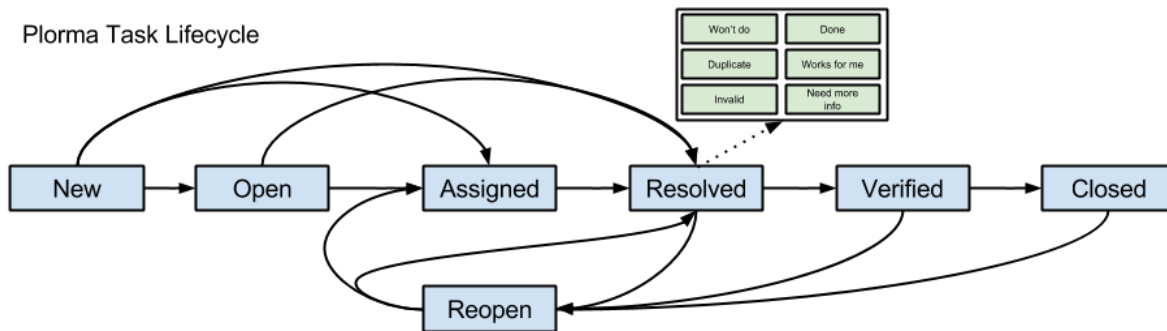| Sever-ity	Description
Blocker	Blocks further development and/or testing work
Critical	Crashes, loss of data (internally, not your edit preview!) in a widely used and important component.
Major	Major loss of function in an important area.
Normal	Default/average
Minor	Minor loss of function, or other problem that does not affect many people or where an easy workaround is present.
Trivial	Cosmetic problem like misspelled words or misaligned text which does not really cause problems

Estimation

The estimate indicates how much work remains to be done until the task is completely resolved. The estimate can be selected from a simplified Fibonacci sequence to regard larger inaccuracy in complex tasks.

The estimate does not have any time unit like hours. It is a abstract estimate and needs to be interpreted individual. The estimate can be used as Story Points in a Scrum development process.

Sprints The sprint listing will show a list of Sprints which are currently in the planning state. You can assign tasks to more than on sprint.

4.1 Lifecycle

4.1.1 States

State	Description
New	Initial state for all new created tasks. Nobody has looked into the task nor it has been checked to be a valid.
Open	The tasks has been checked to be valid. However the task has not been assigned to someone yet. But based on its *Weight of a task* it is queued to be worked on.
As-signed	The tasks has been assigned to a developer. He will start to work on the task based on its priority.
Re-solved	Work on the task has been finished with on of the possible. Resolved tasks may need some QA are acceptance tests. *Resolutions*.
Veri-fied	The resolution has been accepted by the QA. Last steps can be made to finally close the task. Verifying the solution of a task will set the remaining estimate to 0.
Closed	The final state of a task. The task has been resolved the QA has approved the resolution. The resolution has been communicated to all relevant parties. Closing a task will set the remaining estimate to 0.
Re-open	Indicates that an issue has been reopened for some reason. This my be a failed QA or later upcoming issues with the solution. Reopening the task will set the estimate to a unknown value to enforce the user to set a new value for the estimate.

4.1.2 Resolutions

Resolution	Description
Done	Task is done and is ready for QA.
Works for me	Can not reproduce the defect or issue. Everything works as expected.
Need more info	It is unclear what exactly to do here. More information is needed before the work can continue here.
Won't do	Task will not be resolved for any reason.
Duplicate	Task is duplicate of another task.
Invalid	Task is invalid and will not be done for any other reason the formed named resolutions.

4.1.3 Tasks currently under "Test"

You may think that the lifecycle of a task is missing the explicit state that the task is currently under test by someone. Well, of course Plorma provides a way to indicate that a task is currently under test.

Plorma differs between two "states" of the state *done*. If a task is marked as *done* the assignee will be removed automatically. This is because the origin assignee has decided that the task is either finished or the work can not proceed for any other reason. However in this situation the origin assignee is considered not to be responsible for the task anymore. (This is probably what the assignee actually thinks when marking the task as resolved) This is the first state: A task which is done and has no assignee. The task is waiting for someone who will pickup the task e.g for QA.

The way Plorma marks a task to be currently under test is to set a new assignee to the task. A resolved task which is assigned to someone means that his person will do whatever is needed to make the task proceed into the next state (verfied, closed). This can be doing the QA but might also be something different.

4.2 Weight of a task

The prioritization of the task is calculated based on its Priority (think of importance) and its severity. The calculated value is called the *Taskweight*. The Taskweight is used in the task overview are prioritization criteria.

If either the priority or the severity is not set, than the weight can not be calculated and is unknown.

Sprint

Plorma   Home   Tasks   **Sprints**   👤 Peter Product Owner (peter) ▾

Sprint: Winter Sprint (24.11.15 - 31.12.15)

Outline	
Sprint	
Planning	**2**

Planning

Strength ℹ

[   ]

⚠ This field is desired. Please provide a value

Initial Story Points

[   ]

Remaining Story Points

| 0 |

Velocity

| None |

State

| Planning ▾ |

Sprint Backlog

Suchen [   ]   1 bis 7 von 7 Einträgen

+ New

Title	Tags	Priority	Estimate
☐ Fix missing link on "home" button	Defect	70	1
☐ Add multicolumn sorting to overviews	Feature	35	8
☐ Userstory "Bill writes a invoice" does not match anymore	Change	29	1
☐ Data submission fails when user is not registered	Defect	25	1
☐ Add missing "name" field into the export form	Feature	24	40
☐ Call Susan and clarify user story for data import	Question	17	5
☐ Add new ACME export format	Feature , Question		1

⚠ This field is desired. Please provide a value

Submit

Title   The title is used to identify the sprint in various overviews and thus should be concise.

Duration   You need to define the start and end date of the sprint. There is currently no automatism to finish a sprint after the end date has been reached.

Description   Optional. You can give some information on the goals and the purpose of the sprint. Name your expectations and the most important business value which should be reached with the sprint. This information can help the team to keep focused on the most important tasks in the sprint.

Strength   Abstract value which describes the strength of the sprint team. This is usually the sum of time the team

will spent on this sprint. The Strength has no unit but often reflects hours or days. The strength is later used to calculate the velocity of the sprint.

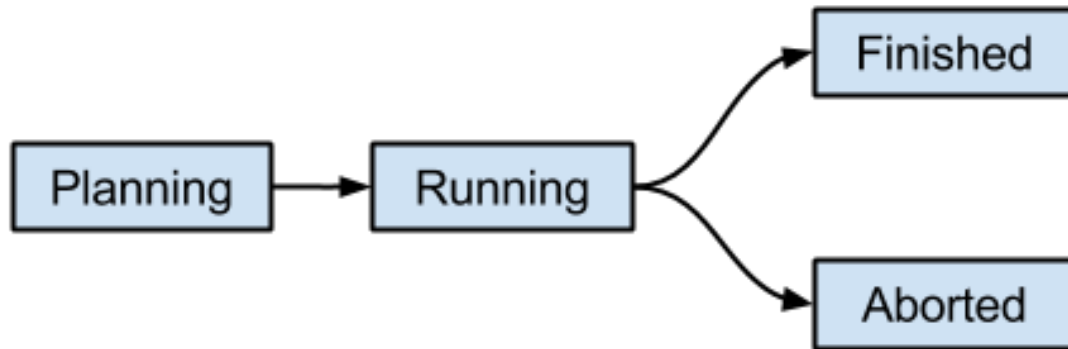**Estimate** This info field sums up the remaining Story Points in this sprint.

Initial Story Points This field will show the sum of Story Point of the initial tasks in the sprint. The value will be calculated when changing from *planning* to *running* state of the sprint. The value is used to calculate the progress in the sprint.

State The sprint can be in one state of the *Lifecyle*

Backlog The Backlog show all assigned tasks to this sprint. While the Sprint is in *planning* state the Backlog lists all open tasks. The tasks are ordered by the *Weight of a task* so the most important tasks are listed.
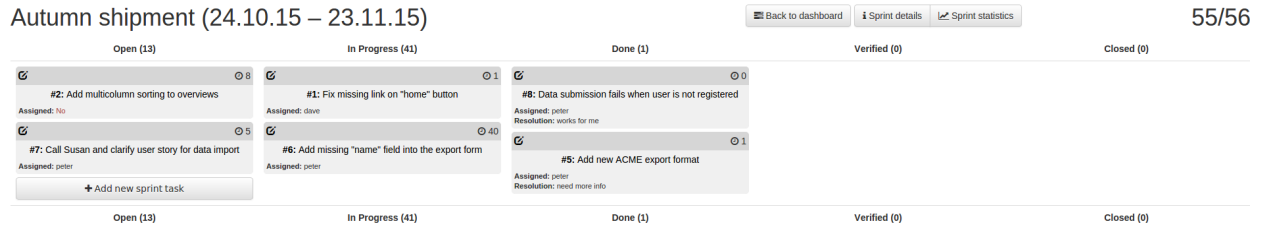
5.1 Lifecyle

Plorma Sprint Lifecycle

State	Description
Planning	The Sprint is not visible to other users in the *Sprints* section of the *Dashboard*. All field are editable. The Backlog will show all open tasks. Tasks can be added or removed.
Running	The start the sprint you need to provide the strength of the team and assign at least on task to the Sprint Backlog. In this state the Sprint is visible to other users in the *Sprints* section of the *Dashboard*. Only the state field is editable. Tasks can not be added or removed.
Finished	The sprint has been terminated in a normal way after the end date has been over. The sprint will not be listed to other users anymore.
Aborted	The sprint has been aborted by the Productowner for any reasons. The sprint will not be listed to other users anymore.

5.2 Sprint Board

The Sprint Board is the major tool while working on the sprint. It gives an detailed overview of the current state of the sprint and allows to quickly document the progress in the sprint or add new items to the sprint. In the right top corner you can see the **total remaining Story Points**.

The Sprint Board is basically a Kanban Board with the following columns. Each column show **remaining Story Points per state**. The states are ordered by the *Weight of a task*.

Open Lists all open tasks. Open are tasks with one of the following the *States*: *new*, *open* or *reopen*.
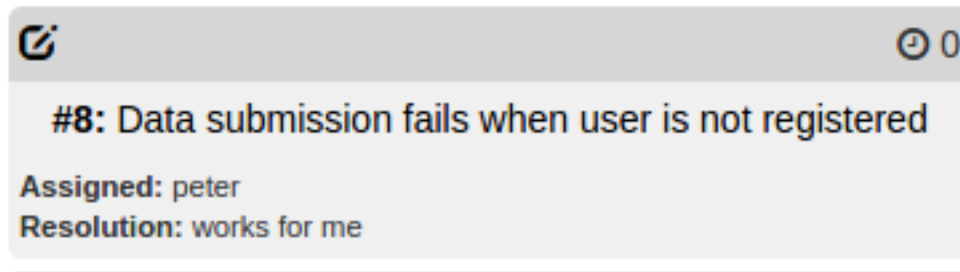
In Progress Lists all tasks which are in the *assigned* state. That means that someone is actually working on this task.

Resolved Resolved tasks are tasks on which the work has been stopped for different reasons. The one who worked on the issue before stopped working on the issue. This can be either because the task is really finished or needs some input or is waiting. The reason why a task has been resolved is the *Resolutions*. However: **Resolved tasks are not done! They need further work or at least a decision on how to proceed with them**.

Verified The resolution of a resolved task has been verified. It is very likely that the task will pass final QA and meets the Definition of Done. It is accepted by the team as a valid solution. The task is almost done.

Closed The task is finally done. No work left. The tasks has passed final QA (e.g Testpushes) and meets all points of the Definition of Done.

5.2.1 Taskcards

Each Taskcard represents a single task in the current sprint. The Taskcards provide minimal required information in the context of the current state of the task. So if a task is in the *resolved* state the Taskcard will also provide the *resolution* of the task. If the task is missing important aspects like and assignee or an estimation than the missing value is indicated by red color.

Clicking on the title of the Task will open a reduced form to set basic attributes of the task. You can add comments, move the task in a different state or adapt the estimation of the task. After saving the task you will go back to the Sprint Board. This way you can quickly document the progress in the sprint and optimize the transparency in the process. By clicking on the edit icon in the top left corner you will get a detailed form of the task.

Indices and tables

- genindex
- modindex
- search

A

C

D

E

K

P

R

S

T

W