



# QGIS PSC Meeting 3 October 2016

---

## Present:

- Tim
- Richard
- Anita
- Gary
- Jürgen
- Andreas

## Apologies:

- Paolo (will join later)
- Marco (will join later)

## Video / Audio Channel

- 

## Previous meeting minutes:

<https://drive.google.com/open?id=1iGxx7N587CxGiXGIVlvrBFW7tVXltY4XXbVogWYJbsg>

## Next meeting:

## Agenda:

- **All: Roundtable:** activity updates (3 minute report back per person)
- Revision of the QGIS.ORG charter (Tim and Andreas made a proposal this week).  
See  
[https://docs.google.com/document/d/1FcChHYgoI4FIURrZRYBjD8kB6IBW\\_BaG0Ieg8TDT0pM/edit](https://docs.google.com/document/d/1FcChHYgoI4FIURrZRYBjD8kB6IBW_BaG0Ieg8TDT0pM/edit)
- QGIS Grants voting results
- Preparing the 2016 annual general meeting (required by law) - this hasn't been prepared yet, and can probably be prepared/discussed by e-mail
- Redmine: 152 projects: move all to issues.qgis.org, and cleanup there? (RD)

- Richard proposes that we keep the legacy projects on the new server and then remove them one by one.
- We will repoint hub.qgis.org to new issues server.
- Release 2.18 LasPalmas, 21/10. Visual Changelog,
  - **TODO:** Tim to send mail out asking for help with the 2.18 changelog
    - 2.18.x will add dwg import
    - 2.18.2 will add improved geopackage
  - **TODO:** Anita to make splash screen with maps from Pablo
- Strategy for publishing 2 versions of plugins (2.x + 3)
- Logo?
  - Working group is busy short listing options
  - Voting members will be given 3 options to choose from

## Roundtable

### Andreas:

- Statutes docs

### Marco:

- 

### Paolo:

- 

### Richard:

- Nothing special

### Tim:

- Writing new training materials
- Working on Org Diagram
- Working on Grants
- New Docker QGIS Server build

### Anita:

- 3.0 Logo working group
- In contact with Pablo concerning map for 2.18 splash screen

### Jürgen

- 2.14.7 / 2.16.3 point releases
- experimental master package repository (stretch/xenial/unstable) - with Qt5 & Python 3 support

**Gary:**

- Porting plugin builder to Python 3 (done)
- New 2x version of the plugin builder (gets rid of QtWebKit)
- How can we publish V3 plugins

## Notes

- **TODO Tim:** Ask Lene about next venue for Spring Hackfest
- **TODO Anita:** 2.18 splash screen

## Revision of the QGIS.ORG charter

Next steps:

- **TODO Tim: Announce general annual meeting within the next two weeks**

## Grant proposal voting results

The PSC has reviewed the process used for voting and selecting the successful grant applications as outlined below:

- Tim +1
- Andreas +1
- Jürgen +1
- Richard +1
- Anita +1
- Marco +1

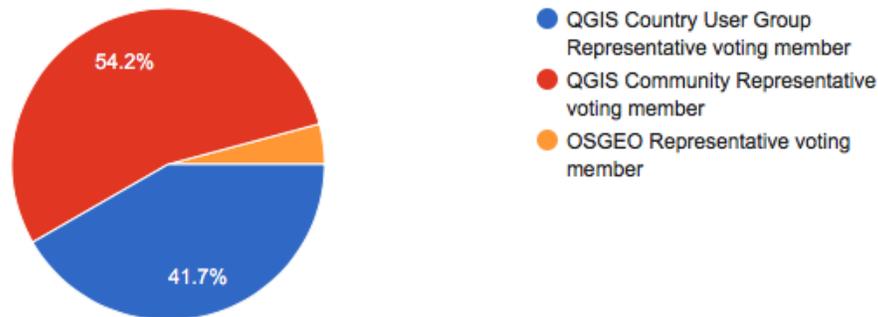
### **Tim to add a list of who did and didn't vote!**

- We had one request to keep the voting anonymous. At first I agreed, but in retrospect I think it is better to keep everything open.
- The full list of votes are available [here](#) on the first sheet. The second sheet contains my calcs for the winner as shown below:
-

Value	Proposals that received votes	Vote count per choice rank						
		1st choice	2nd choice	3rd choice	4th	5th	6th	
€4,500.00	Implement an inbuilt Task Manager in QGIS for background long running tasks	5	10	12	13	16	17	
€2,500.00	Integration of GMT with QGIS Processing as an algorithm provider	1	2	2	2	2	5	
€6,000.00	Introduce everything necessary for QGIS3 to OSGeo4W	6	10	15	18	19	21	
€5,000.00	Project / Map layer registry refactoring	1	4	7	12	14	18	
€2,000.00	PyQGIS Cookbook Review	0	1	1	3	7	9	
€7,000.00	QGIS 3 ticket handling and API refactoring	2	3	6	7	11	13	
€5,000.00	2.16 Documentation	2	3	5	6	10	14	
€10,000.00	Implement a flexible properties framework in QGIS	7	10	13	16	17	17	
€10,000.00	Advanced Toolbox for QGIS	0	3	4	5	5	5	
€460.00	Correction of QgsOgrProvider implementation of GDAL 2.0	0	1	4	6	8	12	
€3,000.00	Explanatory Spatial Data Analysis (ESDA) plugin for QGIS	0	0	0	1	1	1	
€7,000.00	PyQGIS Developer Cookbook update and maintenance	0	0	0	2	5	7	
		€10,000.00	€6,000.00	€4,500.00				
		<b>Total</b>	<b>€20,500.00</b>					
	<b>Methodology:</b>							
	<b>First choice:</b> Proposal with the maximum number of first choice votes							
	<b>Second choice:</b> Proposal with largest sum of first choice and second choice votes wins. In the event of a tie, proposal with more first choice votes wins.							
	<b>Third choice:</b> Proposal with largest sum of first, second and third choice votes wins. In the event of a tie, proposal with more first and second choice votes wins.							

For interest, voting representation was as follows (from 27 potential voters):

## Your voting representation (24 responses)



## Details of the approved grant proposals

### Implement a flexible properties framework in QGIS (Nyall Dawson) - €10,000

**Details:** I am applying for a QGIS grant to cover the implementation of a flexible "properties framework" for QGIS. I honestly believe that implementation of this framework will unlock cartographic power in QGIS well beyond anything that is currently possible in any of the desktop or web based mapping applications.

I propose to implement a system of managing and evaluating properties for generic objects within QGIS. Properties include all settings relating to symbology, such as a line marker's width, color, or offset, label settings (eg font size, color, shadow opacity, etc), diagram properties (colors, size, etc) and composer item settings (position, rotation, frame size and color, etc). While currently many of the properties can be set to use "data defined overrides", the properties framework will extend these capabilities by making them both more flexible and easier to use.

This proposal is being driven by a number of factors:

1. To avoid the current multiple duplicate code paths involving storage, retrieval and evaluation of data defined properties and to make it easier to add data defined support to more things (eg diagrams) without incurring even more duplicate code. Currently labeling, symbology and composer all have their own methods for handling data defined

properties, which makes maintenance of data defined code very difficult.

2. To allow creation of other property types besides the current "data defined" (ie bound to field value or expression result) property, eg time based properties for a future in-built animation framework.

3. To avoid the complexity of requiring users to write their own expressions to map values to colors, sizes, etc and apply scaling functions to these, and instead expose these to users in an interactive, flexible way. Think Mapbox studio's approach to zoom level styling

(<https://cloud.githubusercontent.com/assets/1829991/17850412/6a0f285e-68a0-11e6-8719-cdf74afd061d.jpg>), but available for all property types. Eg data defined values can be set to preset ease in/ease out curves, or manually edited curves through an interactive GUI.

4. Enable the possibility of having live project wide colors. Ie a color palette could be created in the project properties, and color based properties "bound" to these colors. Altering the color would then automatically update every property which was bound to this preset color. This also brings the possibility of "color themes" for maps, eg binding properties to a predefined color types such as "highlights", "background features", etc, and then interactively changing all these color bound properties by applying a color theme to the project.

5. To allow a system of inherited and overridden properties. Eg QGIS default label font overridden by a project default font and finally overridden by label font setting. The proposed composer rewrite (layouts work) would use this property inheritance to bind layout item properties to a dynamic template. Changes in the template would be reflected in all linked layouts, but individual items could overwrite the inherited properties as required. Layout item properties could then be set globally (eg, font size), per project (eg font family), via a "master template" and finally individually per layout item.

6. The labelling engine has a need for predefined label styles. Label properties could be set globally, per project, via a predefined style, or overridden for a particular layer.

Technical details regarding this proposal are available in QEP 22 (<https://github.com/qgis/QGIS-Enhancement-Proposals/issues/38>).

I am seeking funding to:

1. Implement the core functionality for the properties framework
2. Port symbology, labeling and diagrams to the framework, and enable data definable control of all appropriate diagram settings (currently diagrams have a very limited data defined control available)
3. Implement the GUI for the property framework, including:
  - a widget for controlling property behaviour
  - interactive widgets for size and color properties (which have been designed to work inside 2.16's live layer styling dock)
  - interactive widgets for setting the "easing" for properties, with choices of preset ease in/out methods + an interactive curve editor for manual control

If funds are remaining following these items, I will undertake (in order of priority):

4. Bound project colors
5. Begin work on labeling styles

**History:** Because I believe so firmly that this framework is required within QGIS, I have been building toward this work through numerous hours of development over the previous 2 years of QGIS releases. There were a number of prerequisite changes required first, such as the implementation of expression contexts. An initial PR (<https://github.com/qgis/QGIS/pull/2857>) for the properties framework was filed in May 2016, which includes some of the core parts of this proposal. Changes were required based on feedback from that PR, however to date all work on this has been on a volunteer, unsponsored basis and unfortunately I am no longer able to complete such large scale changes as are required by this proposal without funding. Aside from the changes required from the initial PR, significant work remains in implementing GUI, unit tests, and porting symbology and labeling to the new framework.

**Qualifications:** I have an extensive history of large-scale contributions to QGIS since 2013 and a proven track record for writing polished UI with extensive unit testing. I'm passionate about QGIS, being a daily GIS user and strongly believe that this framework is required to take QGIS to the next level of cartographic abilities.

**Implementation Plan:** Due to the extensive refactoring and API changes which are required for implementing the properties framework, this work **MUST** be done in the QGIS 3.0 timeline. If it is not completed during the 3.0 API break period, the amount of work and cost required would substantially increase, and numerous methods across the

symbology, labeling and diagrams API would be deprecated. Accordingly this work will be conducted during the QGIS 3.0 timeline, and for greatest testing I would aim to complete the work ASAP (likely complete by late October). Due to the changes required this work would NOT be suitable to backporting to the  $\geq 2.18$  branch and will be targeted at QGIS 3.0 only.

**Proposal Link:** A QEP detailing technical implementation is available at: <https://github.com/qgis/QGIS-Enhancement-Proposals/issues/38>, and an initial PR available at <https://github.com/qgis/QGIS/pull/2857>

### **Introduce everything necessary for QGIS3 to OSGeo4W (Jürgen Fischer)- €6,000**

**Details:** For QGIS3 we need packages of Qt5, PyQt5 and Python 3 (including many extensions currently available for Python 2). The goal of this proposal is to introduce all required dependencies to OSGeo4W (32&64bit) that are necessary to build and package QGIS3. The requested amount will cover 60h of work on this.

**History:** I also did the packaging of Qt4, PyQt4 and QGIS <3. I've also already started to build and package Qt 5.7 using Visual C++ 2015.

**Qualifications:** See previous point (or well known history)

**Implementation Plan:** I plan on doing it this in Q4 this year to have it available for the release and I don't expect significant extra effort to support Windows (ie. if the issues are solved on a platform that already has Qt5 and friends available it should also work on Windows).

**Proposal Link:**

### **Implement an inbuilt Task Manager in QGIS for background long running tasks (Nyall Dawson) - €4,500**

**Details:** QGIS requires a centralised, in built task manager to handle background threading of long running analysis tasks. Currently these long running tasks are either conducted while blocking the UI (such as when a snapping index is built for a layer) leading users to conclude that QGIS has frozen, via blocking progress dialogs which prevent interaction with QGIS while the operation proceeds, or via custom threaded implementations. By building a standard framework for handling these long running tasks, we will benefit by:

1. Avoiding UI blocking tasks, allowing users to continue working while the task is completed.
2. Simplify background task threading for plugin, processing algorithm (and core) developers by exposing a simple API for creating and scheduling long running tasks.
3. Benefit from the stabler code which comes as a result of having a single, well tested implementation of background threading rather than multiple custom implementations of this code.
4. We "catch up" to our commercial competitors (ie ArcGIS and MapInfo Professional), who currently have inbuilt background threading of long running tasks already available in their software.

This work was begun in <https://github.com/qgis/QGIS/pull/3004>, however significant changes are still required before the task manager can be merged into QGIS. It is vital that the task manager implementation is rock solid and with a future proof API which addresses our needs for the 3.x release cycle.

Accordingly, this grant proposal covers:

1. Building off the work started in the pull request, first addressing the feedback received from GitHub and from direct conversations with interested stakeholders and stabilising the API.
2. Completion of the unit tests to cover all parts of the framework.
3. Polish the GUI for interacting with running and completed tasks.
4. Writing documentation for the Python cookbook demonstrating how the task manager should be used from Python code.

(Please note that this proposal does not cover porting any existing code (such as processing) across to the new framework.)

**History:** An initial prototype of the work was begun in <https://github.com/qgis/QGIS/pull/3004>

**Qualifications:** I have an extensive history of complex changes to QGIS code, and am currently one of the most active QGIS core developers. I have a track record of implementing stable, heavily unit tested code and supporting code I write for extended periods. I am also a daily user of QGIS as a GIS software application, so am invested in making the software as powerful, stable and easy to use as possible!

**Implementation Plan:** This work would be completed ASAP to allow for lengthy testing prior to the QGIS 3.0 release, and to allow the maximum time possible for developers to adapt their code and plugins to the new task manager interface.

**Proposal Link:** An initial prototype of the work was begun in <https://github.com/qgis/QGIS/pull/3004>, and a video demonstration is available at <https://www.youtube.com/watch?v=7pXBZtWYFJc>