



QGIS Grant Programme Applications

September 2016

Contents

Items are listed in order the applications were received (i.e. the order is not significant).

Proposal	Page
1) 2.16 Documentation (Yves Jacolin) - €5000	1
2) Implement a flexible properties framework in QGIS (Nyll Dawson) - €10,000	2
3) Implement an inbuilt Task Manager in QGIS for background long running tasks (Nyll Dawson) - €4,500	5
4) Prepare and translate QGIS documentation and QGIS.org web in Mongolian language. (Tuul Batbaldan) - €10,000	6
5) CheSpatial Utilities Plugin (Edwin Liava'a)- €10,000	8
6) Contribution on translating QGIS API documentation into Mongolian language and prepare documentation (Tuul Batbaldan) - €10,000	9
7) Preparation of QGIS Land Manager2 plugins Help file and user manual (Tuul Batbaldan) - € 10,000	11
8) Gaeta - Geo Analysis & Terrain Animation (Roberto Angeletti) - €2,000	12

<u>9) Correction of QgsOgrProvider implementaion of GDAL 2.0 (Mark Johnson) - €460</u>	13
<u>10) Remote Sensing and GIS Webinars and Online Tutorials based on QGIS (South African National Space Agency – EO) - €6,500</u>	15
<u>11) Introduce everything necessary for QGIS3 to OSGeo4W (Jürgen Fischer)- €6,000</u>	19
<u>12) Advanced Toolbox for QGIS (Murat Kasar)- €10,000</u>	20
<u>13) Explanatory Spatial Data Analysis (ESDA) plugin for QGIS (Stanly Shaji)- €3,000</u>	21
<u>14) Project / Map layer registry refactoring (Martin Dobias) - €5,000</u>	22
<u>15) PyQGIS Cookbook Review (Martin Dobias)- €2,000</u>	23
<u>16) PyQGIS Developer Cookbook update and maintenance (Germán Carrillo)- €7,000</u>	24
<u>17) Integration of GMT with QGIS Processing as an algorithm provider (Remy Galan) - €2,500</u>	28
<u>18) QGIS 3 ticket handling and API refactoring (Oslandia – Régis Haubourg & Hugo Mercier) €7,000</u>	29

1) 2.16 Documentation (Yves Jacolin) - €5000

Details: My proposal aims to speed up the writing of the 2.16 documentation with 7 days (6 days if VAT is not at 0%). Note that French law applies VAT at 0 % for customers outside France.

Based on my former work for the 2.10, 2.12 and 2.14, 7 days should close around 40 tickets plus proof-reading and merging. I will work at least 2 days per week on 4 weeks. I will focus primarily on the OGC and vector chapters then I will use the last hours to other chapters. If the next hackfest decreases the number of tickets under the 40 tickets then I will work on the 2.18 then 3.0 documentation.

If my grant request is too high, I can work less than 7 days.

History: I won't work from scratch and will send push request to the QGIS Documentation 2.16 branch.

Qualifications: I already worked on previous QGIS release of the documentation from 2.10 to 2.14 and still work on the 2.16 release (and so on).

Implementation Plan: As said in my proposal details, 2 days per week for 4 weeks end of september/october.

Proposal Link:

2) 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 ≥ 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>

3) 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>

4) Prepare and translate QGIS documentation and QGIS.org web in Mongolian language. (Tuul Batbaldan) - €10,000

Details: I would like to work on preparing and translating QGIS documentation and QGIS.org web in Mongolian language that many Mongolians could work with QGIS and read documentation in own language. I work in "Topmap" LLC in Mongolia. The company established in 2005 and work on Land Management and Land Information System in Mongolia. We did several work on Land Inventory over Mongolia and Soil quality assessment, land management, land use planing, Geo database training, and capacity building in Land Management Sector in Mongolia. Currently, we work on

creating Land Information System using Open source software like Postgresql/Postgis, QGIS, Mapserver, and pmapper since 2012. I work in GIZ "Land Management and Fiscal Cadastre" project 2005-2012 as a GIS officer, which is implemented at Agency of Land Affair, Geodesy, and Cartography of Mongolia.

Many Mongolian professionals and students use Open Source Geo spatial software these days. Our main problem is English. We speak Mongolian.

We are developing QGIS plugin called Land Manager. In capital city of Ulaanbaatar, we have nine districts Land Offices and specialists from Agency of Land Affair, Geodesy and Cartography (the main land agency) and in countryside 21 aimag (province) has local land offices, in one Aimag land office consists of 10-20 people in one office, and further subdivision of provinces called soum we have more than 300 land officers work in soum level. One soum has one Land manager. We are conducting training courses on QGIS and developed QGIS Plugin to them. How they could start using QGIS in their daily work, how they register parcel application, create contract and print cadastre map and reports using QGIS. Problem is many people does not know English, if QGIS documentation and QGIS webpage are available in Mongolian language, it will be much easier to them start working and improve them self using Open Source software like QGIS and WebGIS. And QGIS will help in their daily work.

History: Related to this grant proposal we created Help file of QGIS plugin using WinCHM in Mongolian language. This help file was useful to many land managers who does not know QGIS and English. In countryside many people does not familiar with computer. In "Land Management and Fiscal Cadastre" project started translating QGIS GUI into Mongolian (mn) and it needs to update and improve for choosing right and correct words in the software translation in future. There is a huge demand of Open Source Desktop application QGIS for Mongolian government organization, universities and different area specialists like Land Managers, Meteorologist, Hydrologist, Environmental officers, Geologist, university teachers, and in private sector, NGO employees so. We did several training courses in QGIS for specialists from Land administration and prepared QGIS documentation and reports into Mongolian language.

Qualifications: I have finished Post Graduate Diploma from Indian Institute of Remote Sensing in Dehradun, India specialized with RS and GIS, and in 2009 I have finished my M.Sc from Faculty of Geo-Information Science and Earth Observation, Twente University in the Netherlands. I have involved at Engineering department, State University of Agriculture in Darkhan for 4 years and GIZ funded "Land Management and Fiscal Cadastre"project for 7 years as a GIS officer. Currently, I am working in private

company called "Topmap"LLC in Mongolia and my responsibilities are on creation of WebGIS using Mapserver and translation of QGIS plugin using Qt Linguist, translation of project report into English and vice versa.

Implementation Plan: The project started in 2014 September funded by Mongolian Government and it will finish in 2017. Preparing and translating QGIS documentation and QGIS.org web in Mongolian language should be start on October 2016.

Proposal Link: We got Government tender work from <http://www.gazar.gov.mn/> this link is Agency of Land Affair, Geodesy and Cartography of Mongolia under Ministry of Construction and Urban Development of Mongolia ([http://mcud.gov.mn/.](http://mcud.gov.mn/))

5) CheSpatial Utilities Plugin (Edwin Liava'a)- €10,000

Details: CheSpatial develop extensions to QGIS as plugins, specifically for Utilities GIS community i.e. Power, Water and Telecommunication. These plugins provide extra capabilities to the "Utilities" GIS which users can add to their own QGIS installation. They are currently under continual development and will be available for use upon official release by early 2017. Users, or interested parties can contact CheSpatial (info@chespatial.com) if they have any questions.

These plugins include but NOT limited to:

Power Plugin (Under-Development)

The Power Plugin is developed with the Power Utility GIS users in mind, it enables them to use these 5 tools.

1. Standardized symbologies according to technical drawing symbols and standards normally used in one-line-diagrams.
2. Automated HV and LV Lines connector to connect Assets with conductors.
3. Data Cleaner for finding records with no spatial objects
4. Technical and Non-Technical Loss Calculation/Simulation
5. Load Flow Study Analysis, Calculation/Simulation

Water Plugin (Under-Development)

The Water Plugin is developed with the Water Utility GIS users in mind, it enables them to use these 5 tools.

1. Standardized symbologies according to technical drawing symbols and standards

normally used in one-line-diagrams.

2. Automated Pipes connector to connect nodes with pipes.
3. Data Cleaner for finding records with no spatial objects
4. Technical and Non-Technical Loss Calculation/Simulation
5. Water Flow Study Calculation/Simulation

Telco Plugin (Under-Development)

The Telco Plugin is developed with the Telecommunication Utility GIS users in mind, it enables them to use these 5 tools.

1. Standardized symbologies according to technical drawing symbols and standards normally used in one-line-diagrams.
2. Automated Lines connector to connect Assets with conductors.
3. Data Cleaner for finding records with no spatial objects
4. Technical and Non-Technical Loss Calculation/Simulation
5. Usage Study Calculation/Simulation

History: These tools have already been implemented in MapInfo as MapBasic Modules, what I am simply doing is migrating them to QGIS by rewriting them in python.

Qualifications: I have been a GIS Developer for Utilities in the Pacific Islands region for 17+ years. More here <http://www.chespatial.com/about-us-and-projects.html>

Implementation Plan: Planning to complete all plugins by first quarter of 2017 and be QGIS 3 ready. Power Plugin - Dec 2016, Water Plugin - March 2017, Telco Plugin - June 2017

Proposal Link: <http://www.chespatial.com/software-tools.html> and <https://github.com/CheSpatial/CheSpatial>

6) Contribution on translating QGIS API documentation into Mongolian language and prepare documentation (Tuul Batbalдан) - €10,000

Details: We would like to make contribution on translating QGIS API documentation into Mongolian language and prepare documentation that many Mongolian students and professionals could use and work with QGIS reading documentation and using QGIS

API documentation in own language. I work in "Topmap" LLC in Mongolia. The company established in 2005 and work on Land Management and Land Information System in Mongolia. We did several work on Land Inventory over Mongolia and Soil quality assessment, land management, land use planing, Geo database training, and capacity building in Land Management Sector in Mongolia. Currently, we work on creating Land Information System using Open source software like Postgresql/Postgis, QGIS, Mapserver, and pmapper since 2012. I work in GIZ "Land Management and Fiscal Cadastre" project 2005-2012 as a GIS officer, which is implemented under Agency of Land Affair, Geodesy, and Cartography of Mongolia.

Many Mongolian professionals and students use Open Source Geo spatial software these days. Especially, QGIS is one of the famous open source software to them. Our main problem is English. We speak Mongolian. If QGIS training materials and API documentations are available in Mongolian language it will accelerate many people to use and work on QGIS software in their work environment.

We are developing QGIS plugin called Land Manager. In capital city of Ulaanbaatar, we have nine districts Land Offices and plus specialists from Agency of Land Affair, Geodesy and Cartography (the main land authority) and in countryside 21 aimag (province) has local land offices, in one Aimag land office consists of 10-20 people in one office, and further subdivision of provinces called soum we have more than 300 land officers work in soum level. One soum has one Land manager. This example is only field of Land Management. We are conducting training courses on QGIS and developed QGIS Plugin to land manager. Problems are first most of the training materials are in English and in countryside many people does not know English. Now they could start using QGIS in their daily work, they register parcel application, create contract and print cadastre map and reports using QGIS. Problem is many people does not know English, if QGIS documentation and QGIS API are available in Mongolian language, it will be much easier to them start working and improve them self using Open Source software like QGIS. And QGIS use not only in countryside local land offices there will be much demand within students and many field of specialists (like meteorology, geology, and ecology etc..) will help in their daily work.

History: Related to this grant proposal on "Translating QGIS API documentation into Mongolian language and prepare documentation" in Mongolian is very first in Mongolia.

Qualifications: I have finished Post Graduate Diploma from Indian Institute of Remote Sensing in Dehradun, India specialized with RS and GIS, and in 2009 I have finished my M.Sc from Faculty of Geo-Information Science and Earth Observation, Twente

University in the Netherlands. I have involved at Engineering department, State University of Agriculture in Darkhan for 4 years and GIZ funded "Land Management and Fiscal Cadastre" project for 7 years as a GIS officer. Currently, I am working in private company called "Topmap" LLC in Mongolia and my responsibilities are on creation of WebGIS using Mapserver and translation of QGIS plugin using Qt Linguist, translation of project report into English and vice versa.

The company I work the "Topmap" LLC is now doing National Land Information System using Open Source software. Before company did several work on Land Inventory for and Soil Quality Assessment, and land management using open source software in nationwide.

Implementation Plan: If grant proposal selected we would be start working on translation and documentation in October 2016.

Proposal Link: We got Government tender work from <http://www.gazar.gov.mn/> this link is Agency of Land Affair, Geodesy and Cartography of Mongolia under Ministry of Construction and Urban Development of Mongolia ([http://mcud.gov.mn/.](http://mcud.gov.mn/)) and one more in allspatial.sc (one of our consultant company)

7) Preparation of QGIS Land Manager2 plugins Help file and user manual (Tuul Batbalдан) - € 10,000

Details: We would like to work on preparation of QGIS Land Manager2 (LM2) plugins Help file and user manual. QGIS LM2 plugin is for registering land applications, contracts, land fee, tax on land privatization, possess, and use rights and printing cadastre maps, land reports for each parcel. We developed first stage QGIS LM2 plugin and it is used for provinces Land Offices over Mongolia. We do have 21 provinces over Mongolia. Depending on user (land manager) knowledge on computer use and QGIS software we need to prepare QGIS LM2 plugin Help file and user manual in Mongolian language. We are conducting training courses on QGIS and developed QGIS Plugin to land manager. Problems are most of the training materials are in English and in countryside many people does not know English. If Help file and user manual exists it will help many people to work freely on QGIS reading the documentation on their computer screen and paper documents.

History: Related to this grant proposal on "Preparation of QGIS Land Manager2 plugins Help file and user manual" in Mongolian is very first in Mongolia.

Qualifications: I have finished Post Graduate Diploma from Indian Institute of Remote Sensing in Dehradun, India specialized with RS and GIS, and in 2009 I have finished my M.Sc from Faculty of Geo-Information Science and Earth Observation, Twente University in the Netherlands. I have involved at Engineering department, State University of Agriculture in Darkhan for 4 years and GIZ funded "Land Management and Fiscal Cadastre" project for 7 years as a GIS officer. Currently, I am working in private company called "Topmap" LLC in Mongolia and my responsibilities are on creation of WebGIS using Mapserver and translation of QGIS plugin using Qt Linguist, translation of project report into English and vice versa.

The company I work the "Topmap" LLC is now doing National Land Information System using Open Source software. Before company did several work on Land Inventory for and Soil Quality Assessment, and land management using open source software in nationwide.

Implementation Plan: If grant proposal selected we would be start working on translation and documentation in October 2016.

Proposal Link: We got Government tender work from <http://www.gazar.gov.mn/> this link is Agency of Land Affair, Geodesy and Cartography of Mongolia under Ministry of Construction and Urban Development of Mongolia ([http://mcud.gov.mn/.](http://mcud.gov.mn/)) and one more in allspatial.sc (one of our consultant company)

8) Gaeta - Geo Analysis & Terrain Animation (Roberto Angeletti) - €2,000

Details: Developing an evolution of GEarthView plugin, to port the same functions on Cesium 3D globe and Leaflet web page.

History: A starting "Gaeta" plugin exists, but it is very experimental, yet.

Qualifications: I created "GEarthView" plugin and it is at fifth position in "Top Downloads" with a number of 164215. I have another version of "GEarthView", ready, that permits to digitize in 3D.

Implementation Plan: Since december 2016, to exit a "Gaeta" plugin 1.0

Proposal Link:

<http://expportocanoma.blogspot.it/2015/10/gaeta-geo-analysis-terrain-animation.html>

9) Correction of QgsOgrProvider implementaion of GDAL 2.0 (Mark Johnson) - € 460

Details: The present code, based on branch release-2_16, does not implement GDAL 2.0 correctly.

The suggested changes made in <https://hub.qgis.org/issues/12479> were not taken into consideration, causing the incorrect listing of valid layers when running with GDAL 2.0. Although an extensive version checking is being done during compiling, almost no checking is being done during run-time, so that functions that do not exist in gdal 1.* (which may be running) are being called when QGIS is compiled against gdal 2.*. This causes qgis to crash with a 'symbol lookup' error.

This second problem can be resolved by the use of run-time version checking.

Adding to the existing 'QgsOgrProviderUtils' class 3 functions which implements the missing functions OGR_GT_Flatten, OGR_GT_HasZ and OGR_GT_HasM when compiled against or running with gdal 1.* resolves this issue (at least on linux-systems), since the 'symbol lookup' logic seems to only check for functions that are actually being used.

The present code uses at least 11 functions that have been deprecated since gdal 2.0.

The major goals of this proposal would therefore be:

Implement static functions in 'QgsOgrProviderUtils' that presently causes qgis to crash when compiled against gdal 2.* and running with gdal 1.*.

Implement static functions in 'QgsOgrProviderUtils' that only use the deprecated functions when compiled against gdal 2.* and running with gdal 1.*. When running with gdal 2.* the corresponding functions introduced in gdal 2.* will be used. At the moment these functions return the same results, but this could possibly change in the future.

Replace existing direct gdal calls with the created 'QgsOgrProviderUtils' functions.

Implement the proper retrieval of layers with common code for both gdal 1.* and gdal 2.* in QgsOgrProvider::subLayers(), bringing the same results.

Adapting src/app/qgisapp.cpp:QgisApp::askUserForOGRSublayers to return both the layerid (as it does now) and the layername, which gdal 2.* must have to work correctly.

Adapting QgsOgrProvider::open to first attempt to resolve the layername and only when that fails to use the layerid to retrieve the layer in the same way as

QgsOgrProvider::subLayers() does.

Testing the adapted code with qgis versions compiled with GDAL 1.11.2 and GDAL 2.2.0dev and running against the different versions.

History: Based on the coding in src/providers/ogr/ of branch release-2_16, compiled against gdal 2.2.0dev, exploratory tests have been made to determine how QgsOgrProvider reacts when running against gdal 1.11.2, 1.11.5 and 2.2.0dev.

When running with 1.* versions, the qgis application is killed, receiving an 'symbol lookup error', caused by 'undefined symbol' errors from libogrprovider.so when using OGR_GT_Flatten, OGR_GT_HasZ or OGR_GT_HasM.

When running with 2.* versions incorrect results are returned when the table contains more than 1 geometry field. In my major database-sample gdal 1.* returns 79 layers, gdal 2.* returns 59. The gdal autotest sample (spatialite_8), that checks for the correct retrieval of 1 table with 2 geometries and 2 views each containing 1 geometry, returns 4 with gdal 1.* and 3 with gdal 2.*.

The introduction of 'Use "layerid=N" instead of "layername=XYZ" for OGR sublayers' (30.06.2016, <https://hub.qgis.org/issues/15168>) correction, fails when running against gdal 2.* since a table containing more than 1 geometry is considered to be 1 layer – as apposed to gdal 1.* where each geometry field is considered to be 1 layer.

Therefore the use of the index based OGR_DS_GetLayer, since gdal 2.0 deprecated, or the gdal 2.* specific GDALDatasetGetLayer functions cannot be used for this purpose. OGR_DS_GetLayerByName / GDALDatasetGetLayerByName must be used to retrieve a layer correctly.

Qualifications: For many years I have worked with the Spatialite project. During this time I have assisted in the testing of new features as well as analyzing reported problems, leading to corrections of the code. Also for the, still under development, RasterLite2 project of Spatialite, extensive work has been done which as lead to a fine tuning of the project during it's development. Also patches have been offered which will extend the import/export functionality to work in a similar way as GDAL, that will be taken into consideration when the development of RasterLite2 resumes.

I maintain the present Android version of 'libsqlite', which supports the present-day version of Spatialite/RasterLite2 and is used in the Geopapparazzi-Project.

<https://github.com/geopapparazzi/libsqlite-spatialite-android>

I also developed the mbtiles and administration portions of Spatialite-geometries and RasterLite2 Databases support for Geopapparazzi.

In the last years I have worked on changes in gdal (mainly in area of OGRSQLite*) to support writable Spatial-Views, together with preparations for a gdal support of RasterLite2, that has not yet been accepted by the gdal-project. It was during the updating of the needed changes to gdal 2.2.0 (from 2.0.0), which until then worked well with QGIS 2.10, that the problems of the present QgsOgrProvider were noticed.

Implementation Plan: LR 2.18 21.10.2016

Since the use of any internal gdal changes (writable Spatial-Views, RasterLite2) must be viewable in QGIS, the goal is to get this completed as quickly as possible.

Proposal Link:

10) Remote Sensing and GIS Webinars and Online Tutorials based on QGIS (South African National Space Agency – EO) - €6,500

Details: Throughout the years, QGIS has grown significantly and has been enhanced to not only provide sophisticated GIS analysis but also provide integration of other open-source software tools and extensions such as GRASS, OTB toolbox, SAGA, Semi-Automatic Classification Plugin (SCP), which are useful for Remote sensing analysis. However, the adoption of the software has been limited in South African Government Departments/municipalities, Universities and Schools due to, among others; the wide adoption of proprietary software, misconceptions that an open-source software (such as QGIS) does not offer enough operations to train and prepare students for a work environment and most of all, the lack of knowledge on the available tools and utility of the software and skills in using the available tools. Hence, capacity building targeting government/municipality GIS officials, academics, university students, school teachers and learners within the African continent is key in expanding the QGIS user community, while building and enhancing the skills in the subject of GIS and Remote Sensing.

The South African National Space Agency (SANSA) has enabling resources to achieve the above mentioned goal, including a wide variety of satellite data, and human resources to develop training materials (Webinars and Tutorial guides) to demonstrate a variety of QGIS functionalities in performing GIS and Remote Sensing related tasks. SANSA's human capital development ethos is that Webinars and online tutorials guides

on the use of QGIS within GIS and Remote Sensing fields will create a whole new pool of QGIS users, while maintaining and generating even greater interest in the current QGIS user community. Topics of the webinars will be formulated to demonstrate specific QGIS functionalities and tools using satellite and GIS data in various application areas. The duration for each webinar course will at maximum be 4 weeks and all webinars will be offered for free and will be advertised on SANSA website, through social media and OS-GEO mailing list. The webinars limited to a maximum of one hour, will be recorded and made available for watching and/or download on the Fundisa Portal site. A minimum number of 100 participants per webinar is expected and participants who complete the course will be offered certificates. Similarly, the tutorial guides, will address specific skills with clear learning objectives and simple, clear step-by-step guidelines on how to perform certain functions within QGIS using GIS and satellite data. These will be available on the Fundisa Portal site for unlimited viewing and/or downloading and will be updated regularly.

History: Fundisa meaning “to teach” is an initiative by the South African National Space Agency (SANSA) which has grown significantly since its inception in 2006. Fundisa Disk is one of the annual offerings under the umbrella of the Fundisa initiative which seeks to expose South African students and instructors interested in pursuing studies in geo-spatial sciences, to open source and funded data and tools. The disk is made up of an assortment of earth observation data such as, satellite imagery, a variety of open source GIS layers, open source software (mainly QGIS), Digital Elevation Models, and GIS and Remote Sensing tutorial guides demonstrated within QGIS. SANSA Earth Observation directorate’s purpose and vision for the Fundisa initiative is to leverage geo-spatial science research at academic institutions all across South Africa, by packaging open and proprietary satellite imagery and products, knowledge and tools. By providing funded data to students, SANSA envisions to enhance research in earth observation, promote multidisciplinary research and empower tomorrow’s scientists, engineers, and decision makers. In (2013), SANSA EO launched a Fundisa Disk School’s Edition with the purpose of supporting Science, Technology, Engineering and Mathematics (STEM) curricula within South African schools. Fundisa Disk School’s Edition boasts a distribution of subsets of variety of satellite data over areas of interest, open-source software (QGIS), and simplified GIS and Remote Sensing tutorials (in PDF & as MP4 video formats) demonstrated within QGIS software. In addition, SANSA has recognised the need to expand data and resources online, hence the launch of the Fundisa Student Portal (<http://fundisa.sansa.org.za/>). The portal is currently undergoing redevelopment. The main goals of the portal are to 1. Assist research and learning by providing useful information, links and course material in earth observation, 2. Enable communication between stakeholders and our earth observation team and 3. Inspire,

excite and leverage research and learning through web mapping applications and other technologies. Fundisa Student Portal consists of tutorial guides in remote sensing and GIS topics, links to the SANSA Data catalogue and other Optical and Synthetic Aperture Radar (SAR) data custodians, links to download open source software (including QGIS), a community and capabilities of viewing SPOT, Sumbandila-Sat, CBERS indexes and the parameters of each scene, among others.

The current work aims at 1. consolidating the available capacity development and building resources within SANSA such as South African Remote Sensing Atlas (<http://atlas.sansa.org.za/>), Web Mapping Services (<http://products.sansa.org.za/mapApp/index.html>), Short Training Courses, Science Advancement activities, and other resources; 2. Introducing new content, namely; Webinars and Recorded instructional videos and update the existing content of the portal such as Tutorial guides to cater for various needs of a wider audience, 3. Providing single rich, comprehensive and friendly online platform appropriate for schools, universities, government departments and municipalities, and 4. Exploring the possibility of online Fundisa Data distribution.

Qualifications: A team consisting of programmers, scientists, students and interns is divided into groups focusing on different aspect of the project. In this way, programmers are responsible for developing the site and maintenance, while Scientists, Students and Interns are responsible for developing and updating the content and engaging with the GIS and remote sensing user community. The webinars will be organised and delivered by our researchers and technologists specialising in different applications of GIS and Remote Sensing with experience in delivering face-to-face workshops.

Previously, the team has developed Fundisa Portal (<http://fundisa.sansa.org.za/>), South African Remote Sensing Atlas (<http://atlas.sansa.org.za/>) and SANSA Web Mapping Service – a platform for showcasing satellite value added products (<http://products.sansa.org.za/mapApp/index.html>).

The team also has wide experience in delivering courses and workshops, and developing tutorials. Some of the courses offered are listed on SANSA's website: <http://www.sansa.org.za/earthobservation/programmes>

Implementation Plan: 1. Phase 1 – Development of Tutorial guides 01/03/ 2017 30/06/ 2017 (4 months)

- 1.1. Identify QGIS key functionalities for demonstration (Basic – Advanced user levels)
01/03/2017 10/03/2017 (10 Days)
- 1.2. Selection of appropriate applications for demonstrating QGIS functionalities
11/03/2017 20/03/2017 (10 Days)
- 1.3. Tutorial Data collection & collation (satellite images & GIS layers over areas of interest in Africa) 21/03/2017 30/03/2017 (10 Days)
- 1.4. Develop Tutorials 01/04/2017 30/05/2017 (2 months)
- 1.5. Quality assurance and online publishing 01/06/2017 30/6/2017 (1 month)
2. Phase 2 – Development of Webinars 01/07/2017 30/04/2018 (10 Months)
- 2.1. Design webinar registration webpage on Fundisa Portal 01/07/2017 15/07/2017 (15 Days)
- 2.2. Design webinar landing webpage on Fundisa Portal 16/07/2017 31/07/2017 (15 Days)
- 2.3. Identify QGIS key functionalities for demonstration (Basic – Advanced user levels)
01/08/2017 10/08/2017 (10 Days)
- 2.4. Selection of appropriate applications for demonstrating QGIS functionalities
11/08/2017 20/08/2017 (9 Days)
- 2.5. Webinar Data collection & collation (satellite & GIS layers in areas of interest in Africa), 21/08/2017 30/08/2017 (9 Days)
- 2.6. Webinar 1 (Proposed focus: Introduction to QGIS and Basic functionalities, plugins, processing toolbox, etc.) 01/09/2017 31/10/2017 (2 months)
- 2.6.1. Develop presentations and materials
- 2.6.2. Advertise Webinar 1 (SANSa website, Social media- Facebook/Twitter, & OSGEO mailing list)
- 2.6.3. Registrations and selection of participants for Webinar 1
- 2.6.4. Deployment of Webinar 1 (1 hour per week for 4 weeks) 01/10/2017
31/10/2017 (1 of 2 Months)
- 2.7. Webinar 2 (Proposed focus: GIS analysis related functionalities) 01/11/2017
30/12/2017 (2 Months)
- 2.7.1. Develop presentations and materials
- 2.7.2. Advertise Webinar 1 (SANSa website, Social media- Facebook/Twitter, & OSGEO mailing list)
- 2.7.3. Registrations and selection of participants for Webinar 1
- 2.7.4. Deployment of Webinar 1 (1 hour per week for 4 weeks) 01/12/2017
31/12/2017 (1 of 2 months)
- 2.8. Webinar 3 (Proposed focus: Remote Sensing analysis related functionalities)
01/01/2018 28/02/2018 (2 months)
- 2.8.1. Develop presentations and materials
- 2.8.2. Advertise Webinar 1 (SANSa website, Social media- Facebook/Twitter, &

OSGEO mailing list)

2.8.3. Registrations and selection of participants for Webinar 1

2.8.4. Deployment of Webinar 1 (1 hour per week for 4 weeks) 01/02/2018
28/02/2018 (1 of 2 months)

2.9. Webinar 4 (Proposed focus: Remote Sensing & GIS applications within QGIS)
01/03/2018 30/04/2018 (2 Months)

2.9.1. Develop presentations and materials

2.9.2. Advertise Webinar 1 (SANSa website, Social media - Facebook/Twitter, &
OSGEO mailing list)

2.9.3. Registrations and selection of participants for Webinar 1

2.9.4. Deployment of Webinar 1 (1 hour per week for 4 weeks) 01/04/2018
30/04/2018 (1 of 2 months)

Proposal Link:

11) 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:

12) Advanced Toolbox for QGIS (Murat Kasar)- €10,000

Details: We are planning to develop an advanced toolbox for QGIS such as ETGeoWizards for ArcGIS. Most of the tools will be focused on solving earth sciences related problems. However, upon request we can develop variety of tools for any GIS application. The toolbox will also include English instructions of the tools and documentation and example sets where necessary.

History: I've attached the below url in "Proposal Link" section as an example visual results of a tool that is coded in Matlab by myself, with the contribution of some online sources, to solve a problem related to my project that is funded by The Scientific and Technological Research Council of Turkey.

Moreover, as Remote Sensing and Geographical Information Systems Laboratory of Middle East Technical University who is lead by Prof. Dr. Lütfi Süzen, we will be working on developing the toolbox under his supervision with a group consisting of a geological, Murat Kasar(<https://tr.linkedin.com/in/muratkasar>), a geotechnical, Bijan Abgarmi(<https://www.linkedin.com/in/bijan-abgarmi-5142094b>), and a computer engineer, Dr. Deniz İren (<https://nl.linkedin.com/in/deniziren> or <http://www.deniziren.com/cv.html>).

Qualifications: I have graduated from Geological Engineering Department of Middle East Technical University. I'm currently at the end of my Remote Sensing and GIS MSc study about mapping chrysotile(asbestos) in southeastern Turkey region for medical geology purposes. I've been working with GIS for over 3 years now and also have developed coding skills meantime.

As I have mentioned in the History questionnaire, we will be working on the project as a group of three researchers. Coding and toolbox design will be the responsibility of myself and Bijan Abgarmi as we are in the same laboratory every working day. Dr. Deniz İren, who is working as a postdocorate researcher at Vrije Universiteit Amsterdam, will be responsible of testing and optimization of the codes in to QGIS. Problems and solution tools to be generated will be decided with the guidance and supervision of Prof. Dr. Lütfi Süzen (<http://users.metu.edu.tr/suzen/Vita.htm>).

Implementation Plan: Depending on the amount of requests by the QGIS developers for additional tools to be added in the toolbox, we anticipate that we can complete the

toolbox in approximately 3-4 months. Thus, we will most probably be able to complete the toolbox before QGIS 3.0 release.

Proposal Link:

https://drive.google.com/file/d/0B7lzgo_GsfHoc1Fna1lwNGVGR0E/view?usp=sharing

13) Explanatory Spatial Data Analysis (ESDA) plugin for QGIS (Stanly Shaji)- €3,000

Details: Explanatory Spatial Data Analysis identifies a collection of techniques to describe and visualize spatial distributions, highlight atypical locations, discover patterns and suggest forms of spatial instability.

These functionalities are currently available into different GIS software as well as open source Python libraries, but not yet integrated within QGIS.

The proposal consists of an integration of ESDA functionalities into a QGIS Python plugin by exploiting PySAL - ESDA functionalities (<http://pysal.github.io>) starting from Hotspot Analysis with Getis-Ord G_i^* statistics [1].

Some effort as well as interest in PySAL integration within QGIS has been risen by the community as it can be seen from different online sources, e.g.:

<http://planet.qgis.org/planet/tag/pysal>

<https://github.com/sjsrey/pysal-qgis>

<http://osgeo-org.1560.x6.nabble.com/pySAL-td5150862.html>

[1] Getis, A., & Ord, J. K. (1992). The analysis of spatial association by use of distance statistics. *Geographical analysis*, 24(3), 189-206.

History: A first prototype of the plugin has been developed yet. Source code, installation procedure as well as test datasets have been made available on GitHub (https://github.com/stanly3690/HotSpotAnalysis_Plugin).

Qualifications: As a Masters Student in Informatics, I have been working as a Full Stack Developer in various projects in the field of technology for past 3-4 years. The privilege to work for QGIS gives an opportunity to put together my experience in technologies and personal interest in Open Source movement.

Prior to the proposal, my contribution towards the development of Hotspot Analysis plug-in in QGIS and hands on experience gained being a part of GeoforAll's United

Nations Training & Documentation gives me a competence to put my effort into the proposed project. These latter activities have been carried out in collaboration with the team of Prof. Maria Antonia Brovelli (fulltime professor of GIS at Politecnico di Milano - Sol Katz Award 2015 - contact: maria.brovelli@polimi.it)

CV:

<https://drive.google.com/file/d/0B8JqCVFh3pSENUF0bzZxMEg4Rm8/view?usp=sharing>

Implementation Plan: Starting from the plugin prototype, the first work to do will focus on dependencies reduction by substituting some of the external libraries functionalities with available PyQGIS APIs. At the same time, it will be evaluated the possibility to provide PySAL - ESDA code within the plugin repository, in order to make it independent also from the PySAL installation. The inclusion of other ESDA tools (e.g Moran's I and Geary's C statistics) from the PySAL core library will represent the second stage of the development process. The work will be carried out approximately in 2 - 2.5 months and delivered within the end of January 2017.

Proposal Link:

14) Project / Map layer registry refactoring (Martin Dobias) - €5,000

Details: here have been various issues identified with QgsProject and QgsMapLayerRegistry classes.

There are three tasks in the qgis3.0_api repo directly linked to the issues:

https://github.com/qgis/qgis3.0_api/issues/8

https://github.com/qgis/qgis3.0_api/issues/16

https://github.com/qgis/qgis3.0_api/issues/23

Essentially, there is a need to redesign how projects and layer IDs are handled in order to support multiple projects within QGIS environment. The most obvious benefit would be simplification of QGIS server code and getting it more in line with "ordinary" QGIS environment (QGIS server has its own way of parsing projects to work around the limitation of one project). Another great advantage is to simplify unit testing of project related functionality and cleanup of code related to embedded projects. It would also open doors to new unforeseen functionality such as handling of projects within browser dock widget.

But the most important part is still to clean up the project API while on the way to QGIS 3 when we are allowed to make changes to the API without having to keep backwards compatibility.

Unfortunately I do not have a ready-to-go QEP yet - I have been putting together some designs, but I haven't settled with a definite solution yet. Hopefully I will have a chance to prepare a proper QEP soon.

History: I have investigated the issue before already, looking through the code and finding ways how to improve the internals. This is going to be mainly refactoring: improving API while removing some of the current limitations.

Qualifications: I have done quite a bit of work on QGIS libraries before and I still strive to keep improving the API so that developing for QGIS is a fun experience rather than a frustrating one.

Implementation Plan: If the proposal gets selected, the work would be undertaken during the autumn/winter, with the target to finish by the end of 2016 so that the new API is fully ready before QGIS 3 API gets frozen (with a comfortably long period to change anything if necessary).

Proposal Link:

15) PyQGIS Cookbook Review (Martin Dobias)- €2,000

Details: PyQGIS Cookbook is still one of the main sources of documentation for PyQGIS developers, however it has not received much attention recently. Over time some of the recommendations and code examples are getting slightly outdated with the fast pace of QGIS development. The idea is to spend some time to review the content, fix any problems and add content - there have been lots of new APIs added to QGIS libraries, however there are often not covered in sufficient detail anywhere in the official QGIS documentation.

History: PyQGIS cookbook is in the QGIS documentation repo, so the plan would be to use the existing documentation and just make it better.

Qualifications: I have started the cookbook longer time ago, but somehow failed to keep it updated as the time goes. I would very much like to make the QGIS developer documentation better, to support new devs wanting to write plugins, possibly some day shifting toward doing some core QGIS work! It is however difficult to find funding for this and in the spare time I usually end up going through the endless list of code improvements I would like to do :-)

Implementation Plan: I would first start with the review, fixing bad or outdated sections and examples. Afterwards I would split the remaining time into taking some interesting bits of API which are currently under-documented and adding new content.

The cookbook would still stick to QGIS 2.x API (QGIS 3 API is not yet finished and will be only picked by developers probably in a year's time or later).

Proposal Link:

16) PyQGIS Developer Cookbook update and maintenance (Germán Carrillo)- €7,000

Details: (Note: You can find this text in the link I provide at the end, I suggest you to read it there as it has a proper format)

The PyQGIS Cookbook is a very important document for both new PyQGIS users and developers. However, it is not rare to read in forums or other community websites that something could not be found in the PyQGIS Cookbook or that a code snippet taken from there does not work. It can be because the user has no previous experience with Python and cannot resolve basic error messages, but it can also be due to lack of maintenance of the code snippets.

I work with PyQGIS almost on a daily basis, which has allowed me to realize that there are confusing parts in the Python API. Things that simply do not work, things that do not work as expected, as documentation states, etc. Just to name some topics that could get into the PyQGIS Cookbook: how to run QGIS Processing algorithms from standalone PyQGIS scripts (I consider this to be very useful), a more detailed Function Editor explanation and examples, and how to access and make use of plugins' methods from the PyQGIS console.

One example of user misunderstanding is how to run a standalone PyQGIS script on Windows, [which is in fact covered](http://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/intro.html#running-custom-applications), but not clear enough for people learning PyQGIS because we still find [questions about it](<http://gis.stackexchange.com/questions/208490/importerror-dll-load-failed-the-specified-procedure-could-not-be-found>). I think "Hello World!" examples need to be given to users, and such initial step should not represent an obstacle for them but an invitation to keep learning.

I am particularly interested in helping people to get started with PyQGIS, I would like to show new users/devs how simple scripts can make a difference in their workflows or in the QGIS GUI itself. Nonetheless, I can also understand and make improvements to more complex PyQGIS uses. For instance, I have answered questions related to QGIS-Processing that cannot be found in documentation, but require reading QGIS Processing code structure.

I propose to:

- Check every code snippet and, if necessary, make changes to it, e.g., remove deprecated function calls, add comments, add alternatives, among others.
- Check every code snippet and notify QGIS devs (via issue tracker, mailing list, Gitter, etc.) and/or GeoAPI maintainers (direct e-mail) about any issue I find (e.g., non-working method, missing classes/functions, and the like).
- Present self-contained (ready to run) code examples. I think there could be a collapsible button (collapsed by default) that displays fully working code (with necessary imports, variable definitions, using perhaps its own memory layers, etc.). As an alternative, we could just show a link to a Gist page, to a file in a GitHub repository, or any other platform you consider. In principle, there would not be a self-contained code example for each code snippet, but I could aim to have one or two per section.
- Find a place for and document 20 (to say something) of the most-voted PyQGIS answers on GIS.SE (or the ones that QGIS community considers) into the PyQGIS Cookbook.
- Find a place for and document 5 (to say something) blog posts from QGIS devs into the PyQGIS Cookbook (QGIS devs could point me to those resources).
- Complete Spanish translation of the PyQGIS Cookbook (for QGIS v.2.x).
- Keep an eye on API changes for QGIS 3.0 and replace/adjust code snippets with the new Python API.

It is unknown to me how different PyQGIS Cookbook versions are managed. I can maintain:

1. A master version (pointing towards QGIS 3.0), and
2. A '2.x' version that you choose between the latest 2.x release (e.g., 2.16.x) or the latest LTR (2.14.x).

I'm aware of the [API Documentation Guidelines QEP](<https://github.com/qgis/QGIS-Enhancement-Proposals/issues/67>) by Martin Dobias, which I find both necessary and complementary. Even though code snippets would be added to the API docs (according to that QEP), I still think the PyQGIS Cookbook is the documentation resource new users will want to read when starting to learn PyQGIS, and sooner or later will be guided (e.g., by the PyQGIS Cookbook itself) to the API docs.

I am open to discuss about the scope of this proposal. If it is selected, I would like to get feedback about it before starting to work.

History: Back in 2010, Martin Dobias started the PyQGIS Cookbook project (<http://osgeo-org.1560.x6.nabble.com/PyQGIS-cookbook-td4109857.html>) to organize and publish PyQGIS code snippets. It is a heavy task for a core developer to keep such documentation up-to-date, specially due to the dynamic and ever-growing nature of QGIS. For instance, QGIS-core devs publish blog posts explaining how to use their newly introduced classes in Python, which not always get into the PyQGIS Cookbook. On the other hand, the Spanish translation is partial and not 100% complete. It is unknown to me who contributed with the translation.

Therefore, my work will be certainly based on other people's work. I want to contribute to their work in order to make the PyQGIS Cookbook *the* reference for people learning PyQGIS.

Qualifications: I hold a MSc. in Geoinformatics from the University of Münster (Germany), I'm a QGIS contributor (due to bug fixes sent years ago as C++ patches), I'm an active member of GIS StackExchange, particularly in PyQGIS-related questions, I'm one of the administrators of GeoTux (a website to promote and share knowledge on FOSS4G), I'm the author of four QGIS plugins and have contributed to QGIS WPS and Simplify plugins. I enjoy helping people to discover how powerful PyQGIS is (and have the patience for that), I enjoy sharing my work with the community, I enjoy learning from

other people's code (even if it's C++ code). I once even met QGIS core devs at the Linux Hotel in Essen, 2012 (unfortunately couldn't meet Martin nor Anita there). I use QGIS every time I can, even as a platform for a couple of experiments (<http://geotux.tuxfamily.org/index.php/en/component/k2/item/298-navigation-dipole-calculus-gizeitgeist-2012> and <http://blog.52north.org/2012/10/04/full-streaming-wps-near-real-time-geoprocessing-with-wps/>). And finally, and most important, I would love to work for QGIS! (Note: The proposal link presents this section with external links)

Implementation Plan: - Mid-October to mid-December (2 months):

- PyQGIS Cookbook for QGIS v2.x (you would choose between QGIS 2.16.x or 2.14.x):

- Reviewed.
- Updated.
- With self contained code examples.
- With the chosen 20 code snippets from GIS.SE included.
- With the chosen 10 blog posts included.
- Translated into Spanish.

- January, February, and March (3 months):

- PyQGIS Cookbook for QGIS v3.0:
 - Sections and code snippets updated.

Notes:

- I would expect to stop working on the 2.x version by mid-December and start working on the 3.0 (master branch) in January.
- If, for any reason, anything cannot be added to PyQGIS Cookbook for QGIS v.3.0, I will leave a list of missing things, so that after the grant programme, the community (and I'm there) can complete it.
- I would not expect to translate PyQGIS Cookbook for QGIS 3.0 because I will be focused on covering the new API.

Proposal Link: <http://gacarrillor.github.io/>

17) Integration of GMT with QGIS Processing as an algorithm provider (Remy Galan) - €2,500

Details: GMT (Generic Mapping Tools) provides a set of several spatial algorithms that are suitable for some geosciences subjects like surface generations from points suitable for gravity data, generation of points like Focal Mechanisms symbols for earthquakes sources representations, generations of rasters for hillshade representation from topographical surfaces, among others.

Integration of these spatial algorithms in QGIS Processing will bring to the geoscience and GIS communities, a complete set of tools for Mapping Generation and Data Analysis, helping to fill the gap between those communities by providing them a common toolset.

We would like to highlight some examples of the powerful GMT map generation:

Worldwide earthquake activity:

http://gmt.soest.hawaii.edu/doc/5.2.1/_images/example_22.png

Distances from Rome to the world:

http://gmt.soest.hawaii.edu/doc/5.2.1/_images/example_23.png

Surface: http://gmt.soest.hawaii.edu/doc/5.2.1/_images/example_05.png

World languages by continent:

http://gmt.soest.hawaii.edu/doc/5.2.1/_images/example_10.png

3-D perspective mesh plot:

http://gmt.soest.hawaii.edu/doc/5.2.1/_images/example_04.png

Image over topographic map:

http://gmt.soest.hawaii.edu/doc/5.2.1/_images/example_32.png

Animations: (Would require looping over map generation changing parameters)

http://gmt.soest.hawaii.edu/doc/5.2.1/_images/anim_04.gif

History: Our work will be based on the PyQGIS API, QGIS Processing framework, GMT algorithms, and the GMTpy wrapper. We want to help link both worlds, geosciences and GIS and make QGIS a reference tool for a variety of communities around the world.

<http://emolch.github.io/gmtpy/>

<http://gmt.soest.hawaii.edu/>

Qualifications: We are a team of 2 persons with complementary profiles. On the one hand, Remy Galan is a MSc. candidate in Applied Mathematics from the National University in Bogota, Colombia, he has worked on Macroseismic Seismology and Geophysics projects where he becomes an active user of GMT for map production, also has worked as web GIS developer for some government institutions. On the other hand, German Carrillo holds a MSc. in Geoinformatics from the University of Muenster, Germany. He is a QGIS contributor, has published several QGIS plugins, and is an active member of GIS.SE, where he answers mostly PyQGIS-related questions. We think that we both can implement a solution that we simply couldn't implement separately.

Implementation Plan: As we will add a new QGIS Processing algorithm provider, we will be tied to QGIS Processing releases (perhaps for a potential QGIS 2.18 or 2.16.x release). As a transition from 2.x to 3.x is underway, we need to make some considerations. We plan to start with the QGIS 2.x API, because of the early stage of QGIS v.3.x at this point. However, after the grant programme has finished and we have added the algorithm provider to QGIS Processing, we plan to support the migration to QGIS v.3.x on our own, when QGIS v.3.x is released and QGIS Processing migrated to the new API.

Proposal Link: <https://remyalex.github.io/>

18) QGIS 3 ticket handling and API refactoring (Oslandia – Régis Haubourg & Hugo Mercier) €7,000

Details: We, Oslandia Team, are very conscious that QGIS 3 needs a serious rework of its API and that QGIS 3 migration will require lots of efforts and coordination. Currently, many API changes are listed in https://github.com/qgis/qgis3.0_api and need tight coordination between developers. We are also aware that such deep and hidden changes are quite difficult to fund via contracts.

Our proposal is to dedicate time for both Régis Haubourg (user, ex-funder) and Hugo Mercier (core committer) to ticket handling, including testing features and bugfixes, of QGIS 3 project. A part of the time will also be dedicated to handle API refactoring tasks.

That proposal seems to fit well with other Grant applications dealing with API Refactoring.

History: Oslandia team has been involved in QGIS project for years and has two core committers. Hugo Mercier and Vincent Mora Régis Haubourg is involved in QGIS community as a user and funder since 2011 through features likes invert polygon renderer, size varying styling assistants, datadefined widget harmonizing or several bug fixing contracts.

Implementation Plan:

We propose to keep track of all API changes proposed until developers come to an agreement and priorities over all the candidates. Another point is also to review all current trackers issues and QEP's and check if their impact is in sync with API changes repository.

In short, we propose project manager time to QGIS 3 project.