



Grant Programme Applications

May 2018

1. GEarthView plugin porting to QGIS 3	2
2. Add OpenCL support for processing core algs	4
3. Welcome page improvements (QEP 102)	6
4. QGIS 3 Enhancement Georeferencer	7
5. Resurrect Processing "R" Provider as a plugin and maintain	9
6. Increased stability for Processing GUI and External Providers and maintenance of Processing	12
7. Update of QGIS Training Manual	16
8. Charts and drawings on attribute forms	18
9. Improve CAD editing tools (first round)	20
10. Export raster symbology in SLD 1.1 (with extensions as needed)	22
11. Get QGIS server OGC compliant and certified for WFS	25
12. Real-life prototyping of QGIS Platform migration to GitLab	29
13. OSGeo4W updates	34
14. Setting registry	36

1. GEarthView plugin porting to QGIS 3

Roberto Angeletti

€ 2000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

I want to port my GEarthView plugin to QGIS 3 version, but I need of infos and resource and motivation.

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

My project is born in 2013 and reached 287001 downloads since now.

It is used every day by hundred of users around the world.

Many of them write me to have the QGIS 3 version.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I am the developer of GEarthView for QGIS 2

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

end of 2018

Proposal Link (e.g. QEP, blog post, etc.)

<http://exporttocanoma.blogspot.it/2015/01/gearthview-20-plugin-per-qgis.html>

2. Add OpenCL support for processing core algs

Alessandro Pasotti

€ 3500

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

Please see the QEP 121 for a detailed description, I already have a working POC, what I'm asking funds for is to complete the task with:

- polishing the code
- testing on windows (and Mac: but I'll need some help here)
- porting more algs (currently I only have only ported slope and aspect) and make use of OpenCL in other critical parts of the code (for instance: hillshade renderer)
- write tests
- create user documentation

Note that the well know proprietary competitor can make use of the GPU for this kind of computations.

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

AFAIK this is the first attempt to bring the powers of GPU computation in QGIS.

Grass has a conditional switch to enable OpenCL but I don't know exactly which alg can make use of it.

Unfortunately the only Qt-OpenCL library I'm aware of was discontinued ~8 years ago and never reached production grade level.

For this reason I'll start from scratches with OpenCL C++ headers.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I've done some contributions to QGIS in the past.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

Target: 3.4

Proposal Link (e.g. QEP, blog post, etc.)

<https://github.com/qgis/QGIS-Enhancement-Proposals/issues/121>

3. Welcome page improvements (QEP 102)

Alessandro Pasotti

€ 2900

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

See: <https://github.com/qgis/QGIS-Enhancement-Proposals/issues/102>

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

Build upon the current projects page but it will be deeply refactored.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I've done some small contributions to the QGIS project in the past.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

Targeting 3.4 (hopefully: I'm not sure how to prioritize all QGIS related tasks that are probably coming in during the next weeks/months)

Proposal Link (e.g. QEP, blog post, etc.)

<https://github.com/qgis/QGIS-Enhancement-Proposals/issues/102>

4. QGIS 3 Enhancement Georeferencer

Mark Johnson

€ 5000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

Resolvment of known problems

Use of existing QgsVectorLayer logic for the placement of Gcp-Points.

Incorporate a Spatialite based support using the (present) QgsSpatialiteProvider, replacing .points file. For the Spatialite specific use of its internal Gcp-Logic, direct calls to Spatialite library are made.

Full details at:

<https://github.com/mj10777/QGIS/wiki/QGIS-Enhancement:-Georeferencer>

which also contains a link to a Pdf showing, step by step, the usage of the new functionality.

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

Based on present Georeferencer in QGIS 2.18. and QGIS 3.0.

This been ported to QGIS 3. Work on a Spatialite based Georeferencer was started in 2016, based on my needs since I use the Georeferencer heavily and simplifies the work process enormously as compared to the present version.

This was Proposal 7 of the 2017 QGIS Grant Proposals.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

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.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

The project, as such, is ready for submission.

Although based on the present code, structural changes have been made to accommodate the present functionality with the needs of QgsVectorLayer together with the Spatialite specific Gcp-Logic.

Due to the QGIS 3.0 API changes, a back porting to QGIS 2 is not intended.

Proposal Link (e.g. QEP, blog post, etc.)

<https://github.com/mj10777/QGIS/wiki/QGIS-Enhancement:-Georeferencer>

5. Resurrect Processing "R" Provider as a plugin and maintain

Nyall Dawson

€ 2500

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

During the QGIS 3.0 development cycle it was decided that some existing Processing providers with dependencies on 3rd party applications would be removed from the default QGIS install. Instead, these providers would be moved to separate plugin-based providers. While some of the affected providers have seen community members step forward and take over maintenance of these providers (such as the OTB provider), the R stats provider has not seen any community members take ownership and develop a plugin based provider for R scripts. Accordingly, users have no way to run their existing R scripts for QGIS under QGIS 3.0.

This grant proposal covers development of a new plugin-based provider for R scripts for QGIS 3.2 and later. We will:

1. Upgrade the 2.x Processing R provider for the QGIS 3.0 API. Processing core underwent a complete rewrite for 3.0, and accordingly the API used by Processing providers is completely changed from 2.x. The code used for the 2.x R provider needs to be updated to use the 3.0 Processing API.
2. Commit to maintaining the new provider plugin for AT LEAST 12 months, ensuring that it is stable and functional.
3. Investigate ways to improve the provider and QGIS integration with R. We believe that

R is quickly becoming a dominant player in the statistical and data science fields because it offers a fantastic set of stable, advanced and performant statistical (and spatial) algorithms. Greater interaction between QGIS and R will benefit both projects, and ensure that QGIS remains a serious tool for geographic data science in the future!

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

This work will initially build off the existing QGIS 2.18 Processing "R" provider. No work has yet been undertaken to port this provider to 3.0 or move it to a plugin-based provider.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I have a long history with the QGIS project, having authored over 7500 commits for the project. I've been submitting QGIS bug fixes and enhancements on a near-daily basis for the past 3 years. Regarding Processing, I was personally involved in the substantial effort to port Processing to c++ during the QGIS 3.0 development cycle and am accordingly intimately aware of the changes required to upgrade 2.18 Processing providers to the 3.0 API. I also have extensive experience with use of R as a statistical analysis tool.

My contributions to QGIS demonstrate an ongoing commitment to writing stable, maintained code with extensive unit testing to avoid regressions in future releases. I'm committed to writing code which is robust and future proof, and does not resort to any "quick hacks" or shortcuts.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

This work will be undertaken during June/July 2018, and the new R plugin will be made available for QGIS versions 3.2 and later.

Proposal Link (e.g. QEP, blog post, etc.)

6. Increased stability for Processing GUI and External Providers and maintenance of Processing

Nyall Dawson

€ 7500

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

Early in the QGIS 3.0 development cycle I received a grant from the Swiss QGIS User Group to commence work on "refactoring" the 2.18 Processing plugin. This work was very extensive, and highlights included:

- Porting the core Processing infrastructure and selected algorithms to c++ (leading to stability benefits and demonstrated performance boosts up to 4x or more)
- Allowing background execution of Processing algorithms
- Near 100% unit test coverage of core Processing and modeler code.
- Exposing Processing algorithms for re-use by associated projects such as QField and QGIS server.
- Many new Processing features, such as transparent reprojection of layers when required, handling of optional outputs, "data defined" parameter values, amongst dozens of others.

While the Swiss User Group grant enabled commencement of this work, the vast bulk of the work has been a volunteer "labour of love" from myself. (Because I personally believe that QGIS is now VERY close to being a first-class data analysis and ETL tool, easily able to rival the expensive commercial options available from FME and Alteryx!).

I am now seeking funding to continue these improvements and for maintenance of the Processing infrastructure.

Close inspection of bug reports filed since the release of QGIS 3.0 has revealed that while the core of Processing and modeler enjoy very few bugs (and those which have been filed are usually fixed within a matter of days – again by volunteer effort!), the most fragile parts of Processing are now the GUI component and the algorithms which rely on external tools (such as GRASS and SAGA).

Accordingly, I would like funding to stabilise these components of Processing and make them "rock-solid". This work involves:

- Refactoring the GUI components of Processing code, making the API safer and more robust.
- Adding more unit tests for Processing GUI (currently there are very few unit tests protecting Processing GUI code, and the frequent regressions seen in Processing GUI are a reflection of this).
- Tracking bugs filed against Processing algorithms, fixing them, and adding unit tests preventing their re-occurrence.
- Closely watching alternative forms of feedback regarding Processing issues - e.g. mailing lists, StackExchange, Twitter, etc and addressing any discussed issues.

(Additionally the current ongoing work of improving the core Processing functionality and included algorithms will continue with renewed vigour! This includes:

- Reviewing Processing related pull requests
- Identifying opportunities for improving the existing algorithms, porting them to c++, and allowing more input parameters to be "data definable".
- Seeking out gaps in Processing's algorithm coverage and adding fast and robust algorithms which fill these gaps.

- Further extending the flexibility of Processing models)

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

This work will build off the current QGIS Processing infrastructure.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I have a long history with the QGIS project, having authored over 7500 commits for the project. I've been submitting QGIS bug fixes and enhancements on a near-daily basis for the past 3 years. Regarding Processing, I was personally involved in the substantial effort to port Processing to c++ during the QGIS 3.0 development cycle and am accordingly intimately aware of the Processing code and the areas of code requiring attention.

My contributions to QGIS demonstrate an ongoing commitment to writing stable, maintained code with extensive unit testing to avoid regressions in future releases. I'm committed to writing code which is robust and future proof, and does not resort to any "quick hacks" or shortcuts.

I'm passionate about this project! I'm not a "fly-in, fly-out" developer who dumps code and disappears, leaving the maintenance burden for others. I'm here for the long-haul and grants like this make it possible for me to continue investing time and effort into QGIS.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

Maintenance work is ongoing and will continue. Refactoring and new unit test coverage will begin both as part of the 3.2 bug fixing effort and over the subsequent QGIS releases. Processing awesomeness will continue to be present in versions 3.2 and above, and will consistently exceed any and all expectations which reasonable users can ever place on the project!

Proposal Link (e.g. QEP, blog post, etc.)

7. Update of QGIS Training Manual

Matteo Ghetta

€ 5000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

Proposal Details

Among documentation released by QGIS.ORG, the Training Manual aims to hold beginners (and less beginners) by the hand and show them step by step what they could do with QGIS and how to proceed. This assumes that this document is up to date with the latest changes to the tools and procedures it describes. Unfortunately, since its release for 2.0, no subsequent modifications have been made on the docs while on the other side, QGIS has got a lot of changes release after release and a great relooking with QGIS 3 release. Of course, some corrective measures have been taken, but often on an ad hoc basis and not with a view to re-interviewing the document as a whole and the chapters in their relations.

This proposal aims to update the Training manual in order to release with QGIS 3.4 LTR an uptodate step-by-step document.

The complete proposal text is available at this link:

<https://docs.google.com/document/d/1gPN5IKMzWOPGrJyUmzy2CX6JnsnXqmVCQO4xeSG7FLQ/edit?usp=sharing>

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

The Training Manual is a great resource that has not been updated since a long time. As described in the detailed proposal we will keep the existing documentation and update/beautify/edit it without starting from scratch.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

Matteo Ghetta: collaborating with QGIS community since QGIS 2.0 making many enhancements to documentation and code. Winner of the Grant Proposal of 2017 to enhance the Processing Documentation.

Harrissou Sant-anna: I joined QGIS community as translator for QGIS 2.0 and years later am bug reporter, code contributor (mainly for the user interface area) and one of the most active contributor to the QGIS Documentation. Writing the user manual gives me a deep knowledge on QGIS features and would be of help with this challenge. As a teacher and issue report cleaner, I have a clear vision of the needs on a QGIS training manual.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

This proposal aims to update the Training manual in order to release with QGIS 3.4 LTR an uptodate step-by-step document.

Proposal Link (e.g. QEP, blog post, etc.)

<https://docs.google.com/document/d/1gPN5IKMzWOPGrJyUmzy2CX6JnsnXqmVCQO4xeSG7FLQ/edit?usp=sharing>

8. Charts and drawings on attribute forms

Matthias Kuhn

€ 7000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

The QGIS attribute form currently allows displaying and editing the following things:

- Attribute editors (highly configurable)
- Relation editor (for showing children of a feature)

We propose to add a new widget type **Charts and drawings** to these types. With this new type it will be possible to show graphically appealing and interactive items like charts or technical drawings on the attribute form. (See proposal link)

Charts and drawings widgets will be highly configurable using QML (a markup language like HTML; not to be confused with the .qml format for layer styles)

QML is able to show complex shapes, integrate pictures from external sources and programmable with javascript for interactive applications. It's a bit like an HTML page on the attribute form but very well integrated with Qt and quickly evolving.

The QML source code will be editable within the "Attributes Form" configuration on the layers properties page. The current feature's attribute values will be available within the QML view for defining chart values.

A wizard will be integrated that allows for

- selecting several input variables from the feature attributes
- a chart type

- a style
 - possibly other visual properties like if a legend should be shown
- and generate QML code from this.

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

The attribute form has been modularized to allow for different widget types which are not necessarily based on a single attribute. This has been done years ago as one of Matthias Kuhn's first contributions to QGIS.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

We have very experienced and active QGIS developers employed at OPENGIS.ch. In particular the following past projects make us the perfect candidate for this project

- * Implementation of the current widget system and relation system for attribute forms
- * Implementation of QField which is completely based on QML and a complex interactive application

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

If accepted, this will be implemented for QGIS 3.4 (LTR)

Proposal Link (e.g. QEP, blog post, etc.)

<https://gist.github.com/m-kuhn/e530f6e838dbbf608f12f2da39f8a18d#file-charts-and-drawings-attribute-form-md>

9. Improve CAD editing tools (first round)

Loïc Bartoletti

€ 6000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

I propose to start the realization of the QEP #106 by developing two very awaited functionalities for the drawers coming from CAD:

- Reverse curve / linestring (aka swap direction)
- Trim / extend

1 Reverse

A new icon will be added to the edit toolbar. By selecting it, it will be enough to click on a feature to reverse its geometry. For this I will rely on existing "reversed" functions on curved and inherited geometries. This tool will only apply to curves and lines. It will keep the Z and M values.

2 Trim / extend

Among the many possible implementations, I propose in this grant to realize two distinct tools (one to trim, one to extend). They will only apply when there is a strict intersection (First case in

https://raw.githubusercontent.com/lbartoletti/lbartoletti.github.io/master/QEP_snapping/extend_cases.png

<https://github.com/qgis/QGIS-Enhancement-Proposals/issues/106#issuecomment-335959367>).

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

This proposal is the continuation of my past developments on QGIS to improve tools on geometries

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I have already done some geometry plugins in QGis 2, which I ported to QGIS 3 in addition to other geometry algorithms.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

Summer 2018

Proposal Link (e.g. QEP, blog post, etc.)

<https://github.com/qgis/QGIS-Enhancement-Proposals/issues/106>

10. Export raster symbology in SLD 1.1 (with extensions as needed)

Andrea Aime

€ 3000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

The current QGIS 3.x SLD symbology supports point, line, polygon and text symbolizers. The raster one is the last missing symbolizer in the SLD set, this proposal aims at filling this gap using as much as possible standard functionality, and when that one is missing adopt vendor extensions inspired by the GeoTools ecosystem as possible/suitable.

In particular, standard SLD/SE syntax will be used for:

- * band selection
- * single band false color display (continuous or classified)
- * basic contrast stretching
- * basic hillshading (won't support all the options in QGIS)

The following will use extension instead:

- * Contrast stretching with specified min/max
- * Color blending with other layers

Functionality not scheduled to be supported (typically controlled outside of the style, in a dynamic way):

- * Resampling control (nearest neighbour, bilinear, bicubic)

Since the export will be in SLD 1.1, GeoTools/GeoServer might also need to be tweaked/fixed to properly support the syntax, that will be hopefully addressed using

some OSGeo UK funds that the GeoServer PSC has available, creating a solid export path between the two softwares.

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

Andrea has started working on QGIS export with a set of basic fixes during the FOSS4G 2016 code sprint, with the kind support of some QGIS code developers. The work went through a number of reviews and was merged. Eventually it was followed up by scale based symbology, and work on improving export of point symbols and parametric SVGs.

Later, along with GeoSolutions, he picked up the text symbolizer work in progress coded by Raymond Nijssen and build a comprehensive export of label symbology to SLD, which was also reviewed and merged in time for the QGIS 3.0 release.

Both cases came with a set of python tests.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

As the “history” part shows, Andrea has been working on closely related issues before. This work lays the base for the raster symbology export will will be performed in the same code area, and tested in a similar fashion.

The good knowledge of SLD/SE, along with experience with the GeoTools map rendering engine, is also a important factor in this proposal.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

The work is scheduled for implementation during the summer/autumn so that it can be ready for including in release 3.4.0 in October 2018.

Proposal Link (e.g. QEP, blog post, etc.)

11. Get QGIS server OGC compliant and certified for WFS

Régis Haubourg

€ 6000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

In 2017, QGIS server was fully re-factored and we created a continuous integration platform that tests every night OGC compliance with WMS 1.3.0. [0]

This revealed a lot of issues and QGIS.org granted us a budget to reach the WMS 1.3.0 certification.

We now succeed to all the basic, recommended and queriable levels, so we launched the official certification process with the OGC.

Now that we have a platform, and we successfully went through the whole process once, we want to go certify the second most used OGC webservice, WFS!

This will ensure you that whatever the client you use, it will always behave as expected.

We are really excited to have soon QGIS server officially listed as a compliant WMS and WFS server!

****Why do we apply for a grant application ?****

Well, gaining and maintaining certification will build a long term trust for companies or authorities who choose to rely on QGIS server.

Also because we trust in QGIS, as a really powerful, easy to administrate and easy to publish web server.

And finally, because certification process is not something end user will fund easily, this is why grant applications really look like the right way to address such tasks.

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

At Oslandia, Paul Blottiere and Régis Haubourg have been involved in the QGIS server refactor, and we built up an open source platform that embeds the OGC TeamEngine [1] testing tools, QGIS latest server builds, and a Geoserver reference implementation [2] Paul has been spending a lot of efforts in progressively fixing platform issues with TeamEngine - which took some time - and then tackling all OGC failures.

We now have a good knowledge of both the OGC TeamEngine and the standards.

[0] <https://github.com/Oslandia/docker-qgis-ogc-cite>

[1] <http://cite.opengeospatial.org/teamengine/>

[2] http://test.qgis.org/ogc_cite/

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

At Oslandia, Paul, Vincent and Hugo are involved in the QGIS server refactoring as core committers.

Régis is also committed to the QGIS project life and especially in QGIS server coordination.

We are very concerned that this area needs to be dealt with care and seriousness, because many of you rely on it for your production tools.

And we are permanently working in the server code, to make it always leaner, more robust, and faster at every QGIS release.

Oh, and between us, don't tell anyone, we are working on a performance benchmark

testing infrastructure and it seems QGIS 3 is effectively faster than that good old QGIS 2.18 ;-). Would wouldn't like a faster, more reliable, and standard server working seamlessly with your favorite GIS desktop ?

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

Our first task will be to instantiate a new OGC Team engine Docker with WFS services.

We are relying on OGC teamEngine REST API to call the test suite.

This step can take some time because we encountered TeamEngine crashes, and we'll work with OGC to fix those issues upstream.

Then we will make the deploy the platform so that it generates nightly reports.

This step will give us a clear view of the failing tests and the current degree of compliance.

If we get enough time at this step, we'll try to tackle most of the issues and will try to reach the 100% compliance and certification. Some issues might need heavy re factor though, we'll keep that for later work if this occurs.

****Risks and uncertainties: ****

Given that we made it with WMS, We are pretty sure we'll make it. However,we will rely on the CITE TeamEngine tool, and we can't be sure we won't hit some time consuming issues.

Hopefully, OGC platform can let us test several WFS versions via a the REST API. But there is no way to know without to do the real work. So if things go well, and QGIS is

close enough to the standards, we could go really fast there and land fixes before 3.2 !
Of course, we'll try to backport as much fixes as possible to 3.0.

Proposal Link (e.g. QEP, blog post, etc.)

<https://github.com/qgis/QGIS/wiki/OGC-compliance-for-QGIS#qgis-32-plans>

12. Real-life prototyping of QGIS Platform migration to GitLab

Régis Haubourg

€ 6000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

The QGIS project is currently organized around a set of collaborative tools for contributors.

QGIS software and associated subprojects, like documentation, website and more, need to maintain an infrastructure with all their history of development, ease of use for their end-users and developers, modern architecture and an evolutive solution.

QGIS currently mainly uses Redmine and GitHub, for issue management and code hosting and management.

The current situation is far from ideal, as identified through previous discussions :

Using separate tools complicates the workflow

Redmine is considered old software and lacks some features and good ergonomics

GitHub is a proprietary platform

There has been previous discussions about working on a more coherent and modern platform, and the final position was that we need a proof of concept before taking a definite decision on migration. This is a hard and complex task which will impact all

power users and developers, and as such should be taken care of with meticulous and professional attention.

This is exactly what this Grant Application is for: create a complete prototype of full migration from Redmine and GitHub to GitLab, so as to be able to take a migration decision while having proof-tested the proposed solution.

We don't want to lose any information from the current platform, and try not to lose any feature in the process neither. GitLab has proven meeting these requirements, and the Community Edition is OpenSource and actively developped.

For this grant application, we intend to run our own instance, and the server's setup and running costs for a year are included in the proposal.

Most risks have already been identified and we started working on a prototype, but without further funding, we cannot tackle all problems. We therefore require and request a grant application, with confidence in succeeding in our goal. We will setup the platform, migrate the data from Redmine and GitHub, solve every small glitches and problems, and report our experience, with a detailed proposition for an actual migration.

As shown by preliminary work, we realize that this is the good time to launch such an initiative in the life of the QGIS project, and we should not miss this opportunity !

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

This work does not start from scratch and tries to take the most out of the existing following work and discussions:

Discussion on the mailing list and at Madeira's community meeting about migrating away from Redmine

Initial review of possibilities and preliminary work on GitLab assessment

The OpenSource GitLab CE project : <https://gitlab.com>

Redmine to GitLab migrator project :

<https://github.com/redmine-gitlab-migrator/redmine-gitlab-migrator>

The sysadmin work at Oslandia to use GitLab as an internal tool for all projects

The work achieved by Oslandia on building OSGeo4W packages on GitLab-CI infrastructure

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

We have gathered a team to be fully qualified on the subject, and able to tackle all aspects of such a complex migration.

At Oslandia, Régis Haubourg is deeply involved in the QGIS project. As an active community member, he navigates through the thousands of issues, pull requests, reviews and master the current collaborative tools and their limits.

His colleagues Ludovic Delaune, and Paul Blottière are system administration experts. Paul is also a QGIS core committer. They set up Oslandia's GitLab infrastructure, and they are fluent with its administration and features.

Vincent Picavet started working on a proof of concept for GitLab migration of Redmine issues at Madeira's community meeting, and will be able to provide them with an efficient kickstart.

Last but not least, Jocelyn Delalande joins the team. He is the main developer of the Redmine-GitLab-Migrator tool, which does most of the complex magic spells. He also happens to be a QGIS user, and would be glad to be able to help and contribute to this migration effort.

Yes, this is a dream team for this complex project!

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

Implementation plan

We already designed a proposal for the migration plan here:

<https://github.com/qgis/QGIS/wiki/QGIS-Platform-migration-plan#actions-plan>

Our goal for this grant application is not to actually switch to GitLab, since it would need a definite decision from the project, but to positively achieve everything needed to be able to just push a button to activate the migration.

The implementation plan is (in short) the following:

- Set up a specific in-house GitLab instance

- Migrate all users from Redmine and GitHub to GitLab

- Migrate all elements from Redmine to GitLab

- Migrate all elements from GitHub to GitLab

- Set up URL rewriting to keep all external links to current Redmine valid

- Create a GitHub mirror of the GitLab instance and keep Travis-CI happy

- Check the resulting projects for incoherencies or errors

- Open the GitLab instance for the community to test it for a limited time

Write down a summary of the prototype, with known problems and workarounds, as well as a proposal for actual migration plan

"The devil is in the details": a specific attention will be put to little things which could reveal to be blockers for a migration. URL stability for example is important.

As for schedule, the work would span over June and July, so as to be able to test during the summer, and be ready before September. A full migration could therefore be decided before the next stable QGIS release.

Proposal Link (e.g. QEP, blog post, etc.)

<https://github.com/qgis/QGIS/wiki/QGIS-Platform-migration-plan#actions-plan>

13. OSGeo4W updates

Jürgen Fischer

€ 4000

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

I would like to do various pendings updates in OSGeo4W:

- * PROJ 5.0.1 with proj-datumgrid
- * GEOS 3.6.2
- * GDAL 2.3
- * Python 2.7.14 (to revive pip for QGIS LTR)
- * Qt 5.10 (with updated reverse dependencies qca, QtWebKit, QScintilla, qtkeychain, sip, PyQt5, Qwt/PyQwt)
- * gpsbabel
- * possibly more will turn up in the process

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

I've been maintaining the Windows builds of QGIS using OSGeo4W for years.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I've been maintaining the Windows builds of QGIS using/maintaining OSGeo4W for years.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

QGIS 3.2/3.4.

Proposal Link (e.g. QEP, blog post, etc.)

14. Setting registry

Denis Rouzaud

€ 6700

Proposal details *

Provide a detailed description of what it is you would like to work on. Reference QEP's or other online resources if needed.

QGIS application settings are spread over the source code and are prone to many errors (typo, different default values, type change (e.g. int to enum, double to int), etc.).

We propose here to introduce a registry for settings which will allow preventing such errors.

As a note, the registration of settings will not be mandatory, and current class/methods (QgsSettings) will remain. This means that any code related to settings in a plugin will continue to be valid after this development.

This registry will also provide some extra information like description or validity domain (min, max, etc).

As a first step, these informations will allow to greatly improve the advanced settings editor by providing appropriate widget with a text hint (instead of a simple line edit).

Even more, for enum based settings, it will provide a combo box filled with the enum names instead of meaningless integer values.

Second, it will offer an API to associate widget to settings. It can then automatically set the widget from the setting value and later set the setting from the widget value (on value change or dialog acceptance). These features are currently available as a Python library (<https://github.com/3nids/qgissettingmanager>) and has proven to be very useful for plugin authors.

During this process, existing settings from QGIS core will be ported to the new registry. This will be the occasion of bringing more coherence into naming of settings.

We believe this is a great addition both for the correctness and completeness of QGIS core but also as a time saving tool for plugin authors.

History *

Tell us a little about what work has already been done related to your grant proposal. Are you starting from scratch or are do you plan to build on the work of others?

Some work has already been done for enums in QgsSettings and some of the GUI features are already provided as a Python library.

How are you qualified to do this? *

Tell us about previous work you have done that will demonstrate that you have the needed skills and enthusiasm to complete this task...

I am a QGIS core and plugin for a few years now (2012) with a solid experience with application settings both core-wise or as a plugin author.

What is your implementation schedule / plan? *

Lay out for use what will be done when. Please try to tie your work plan to the QGIS release schedule and other key activities in the QGIS project.

if there is no major change in the QEP, this is planned to be achieved for 3.4

Proposal Link (e.g. QEP, blog post, etc.)

<https://github.com/qgis/QGIS-Enhancement-Proposals/issues/124>