DGIWG GitHub report

Table of Contents

[1. Introduction](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_introduction)

[1.1. Context](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_context)

[1.2. Aim](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_aim)

[2. Experimentation description](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_experimentation_description)

[2.1. GitHub setup](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_github_setup)

[2.2. Tutorial](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_tutorial)

[2.3. Document template](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_document_template)

[2.4. Standard development](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_standard_development)

[2.5. and what about GitLab ?](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_and_what_about_gitlab)

[3. Feedback](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_feedback)

[3.1. Advantages](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_advantages)

[3.2. Things to be checked](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_things_to_be_checked)

[3.3. Recommendations and way forward](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_recommendations_and_way_forward)

[Annex A](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_annex_a)

[Annex B](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#_annex_b)

1. Introduction

1.1. Context

OGC is using GitHub tools for developing standards and testbeds reports. This process has several advantages:

* a unique gold copy of the document (and other resources like schemas)
* version control of the resources
* focus on the content and not on the look of the document (also avoid errors in references, text styles, numbering, …​)
* easy contribution and merge of the comments/contributions with full traceability

Associated to asciidoc text format, PDFs or HTMLs can be generated quite easily for document publication.

*PS : the NGA also develops its own profiles on Github (*[*https://github.com/ngageoint/*](https://github.com/ngageoint/)*).*

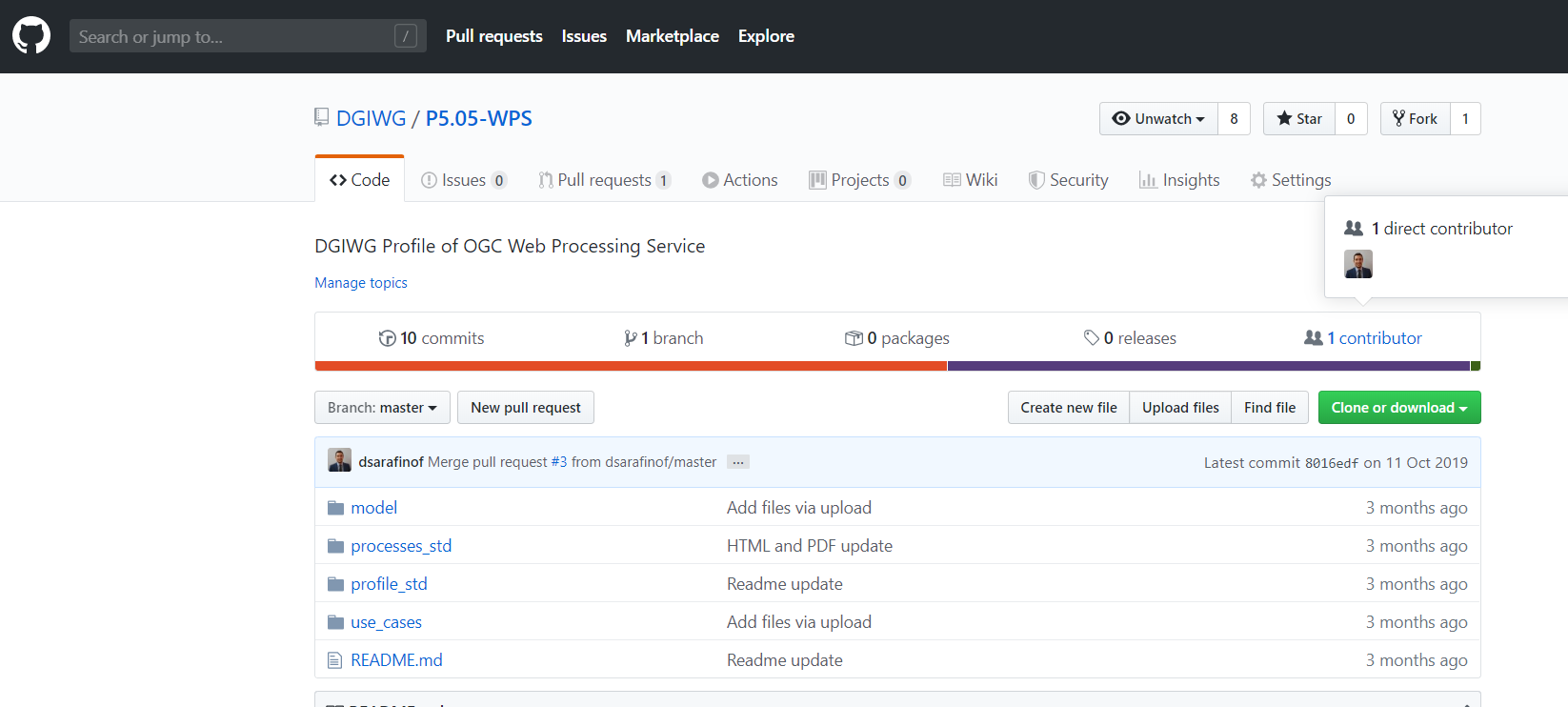
1.2. Aim

DGIWG management team agreed on a one year experiment period for the P5-05 project (Web Processing Services). A DGIWG experimentation has then been run in 2019 to investigate use of GitHub tools for helping the process development of DGIWG standards.

2. Experimentation description

2.1. GitHub setup

DGIWG organization has been created on GitHub website <https://github.com/DGIWG/>. This allows creation of open repositories for DGIWG projects. Concerning the DGIWG experiment, the WPS test has been run under <https://github.com/DGIWG/P5.05-WPS>.



**Figure 1. DGIWG P5.05-WPS repository**

|  |  |
| --- | --- |
| **NOTE** | the experiment has been run on a voluntary base; usual contributions/comments to the document have also been accepted during the process. |

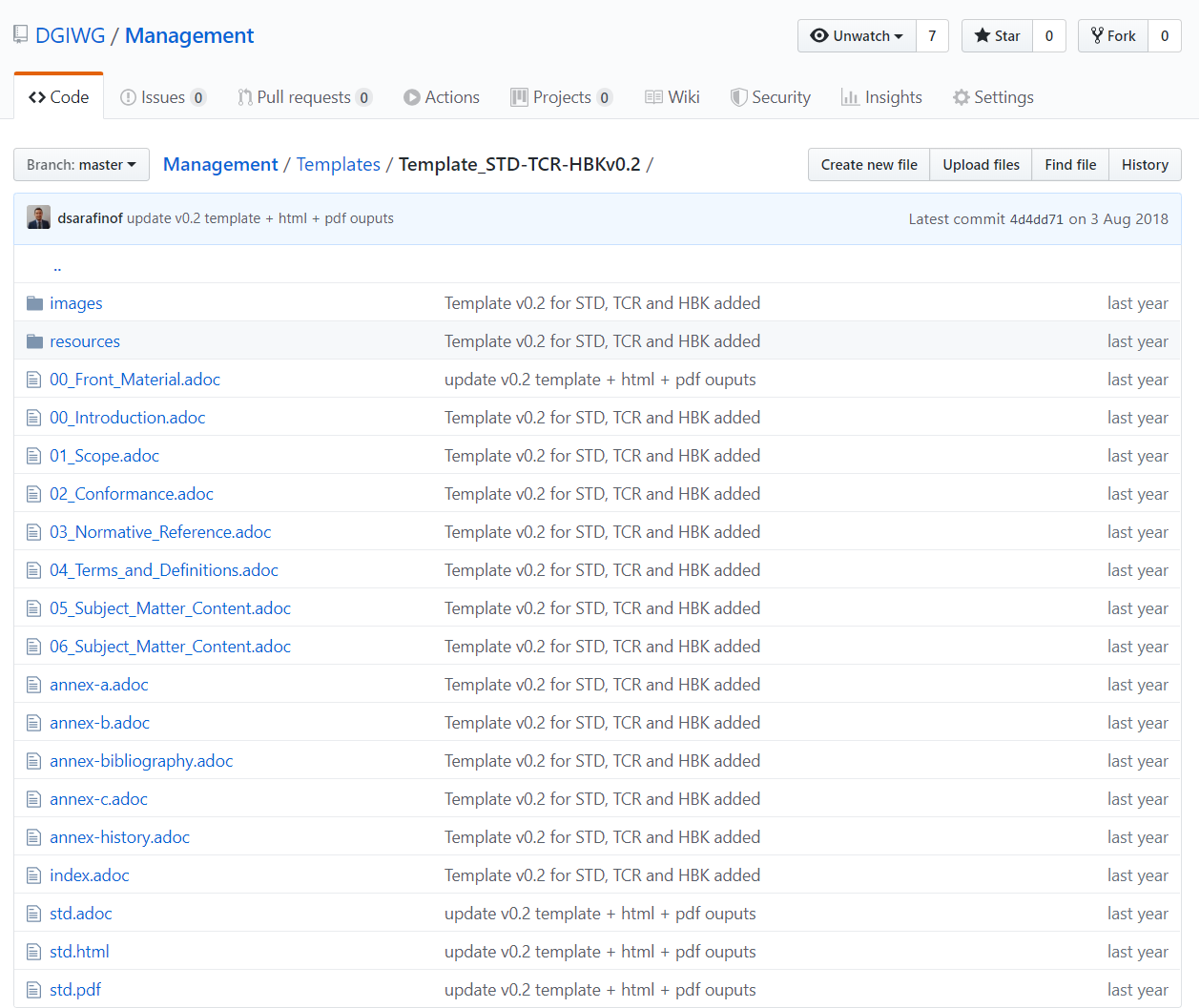
2.2. Tutorial

Git/GitHub tutorial has been created to present to the DGIWG WSTP experts and more globally to DGIWG members during side sessions and DGIWG TP plenary <https://portal.dgiwg.org/files/?artifact_id=69392>.

A test repository (sandbox) has been created to help volunteers to play with it before contributing to the project (<https://github.com/DGIWG/sandbox>). A "step by step" GitHub document has also been created for this purpose (see Annex A); it also describes required and recommended tools to be used.

2.3. Document template

First draft of DGIWG template document has been generated for testing purposes <https://github.com/DGIWG/Management/tree/master/Templates>. This template provides an empty standard document (AsciiDoc format) with same structure and close look and feel as current DGIWG word template for DGIWG standards.

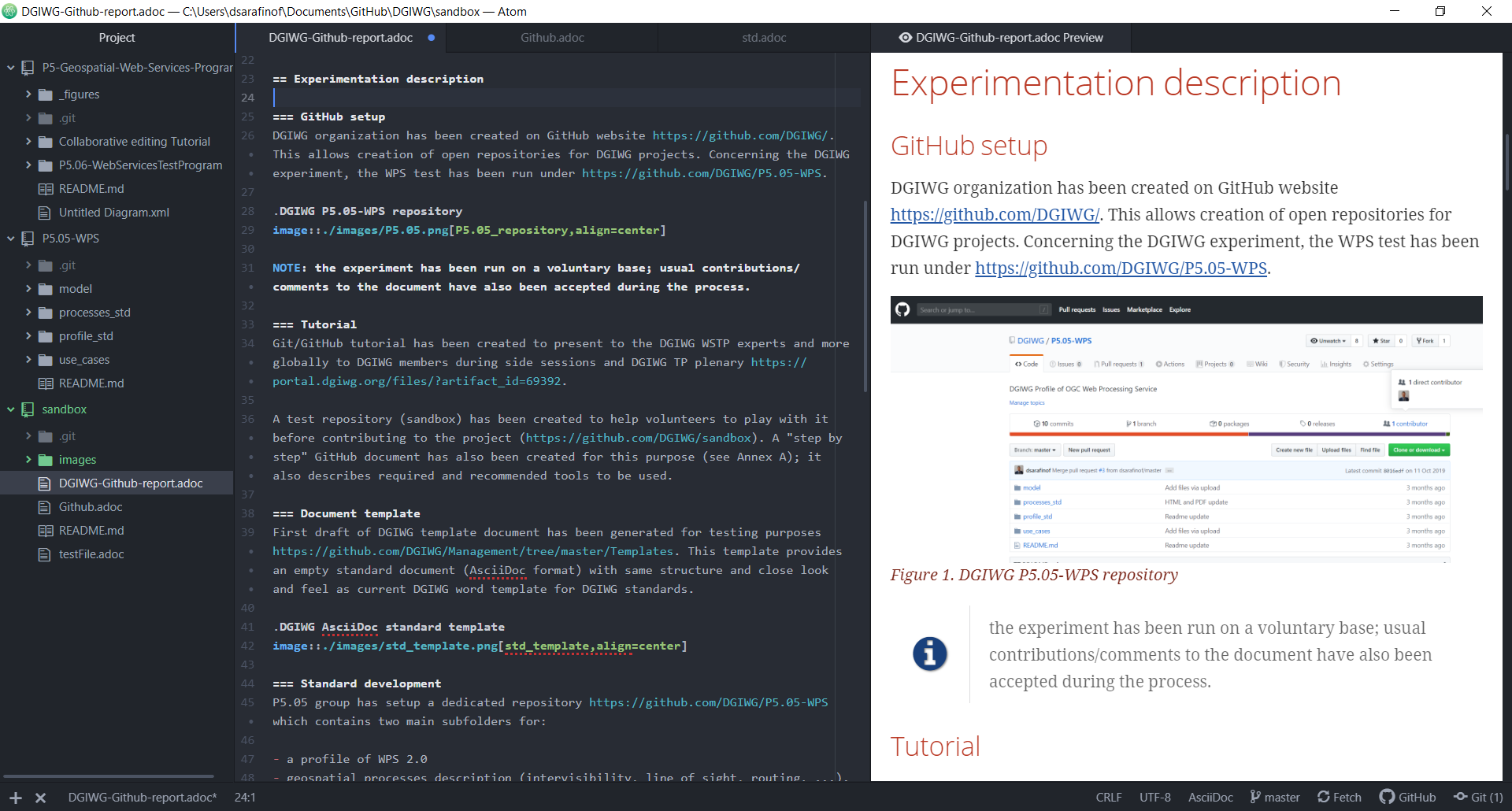


**Figure 2. DGIWG AsciiDoc standard template**

2.4. Standard development

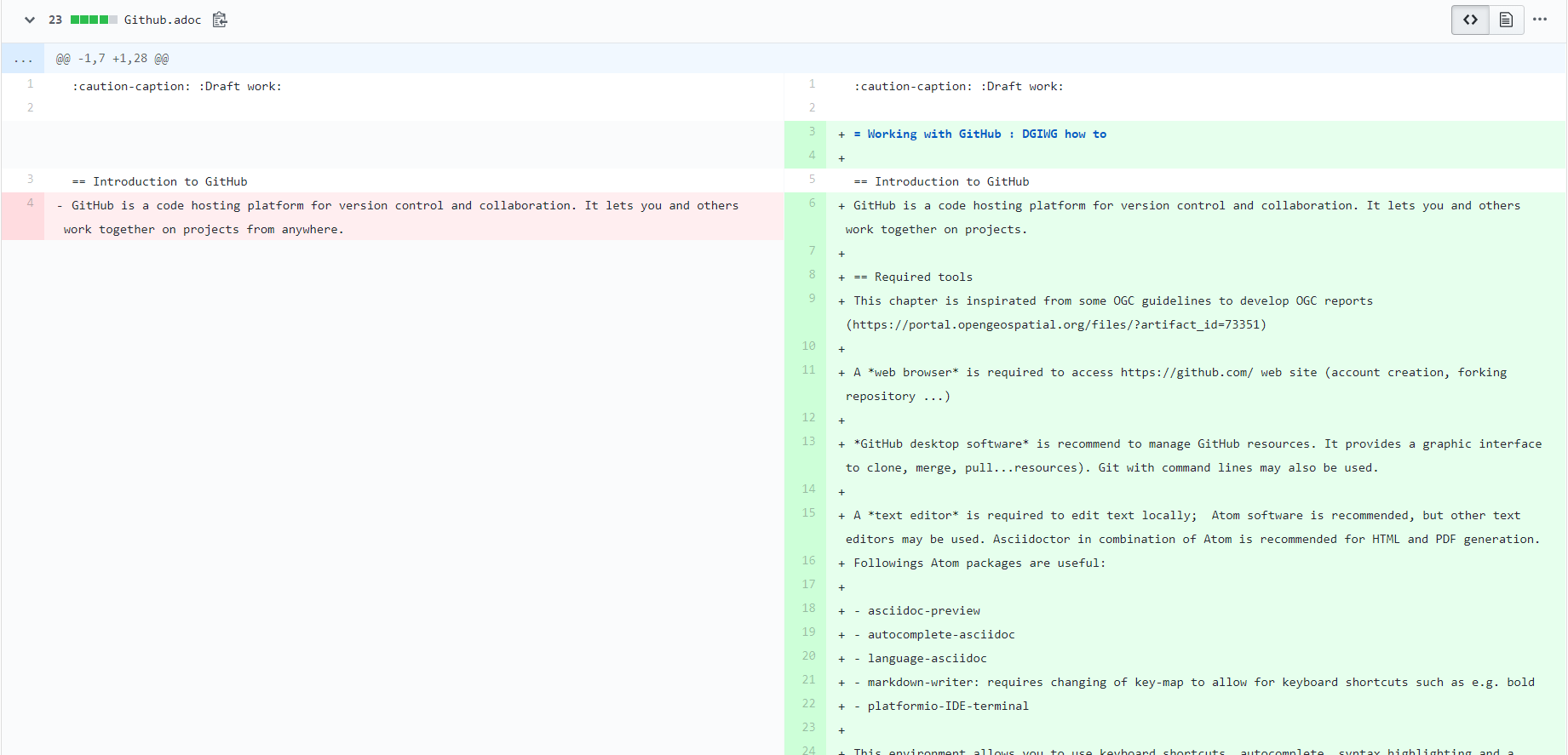
P5.05 group has setup a dedicated repository <https://github.com/DGIWG/P5.05-WPS> which contains two main subfolders for:

* a profile of WPS 2.0
* geospatial processes description (intervisibility, line of sight, routing, …​).



**Figure 3. Edition of documents**

Contributions (with text editor Atom) have been made on each expert clone of the repository before submitting pull request to the DGIWG repository.



**Figure 4. GitHub comparison during the pull request process**

Each of these two documents follow the previous defined template. command lines allow easy generation of PDF or HTML documents.

|  |  |
| --- | --- |
| **WARNING** | insert github workflow |

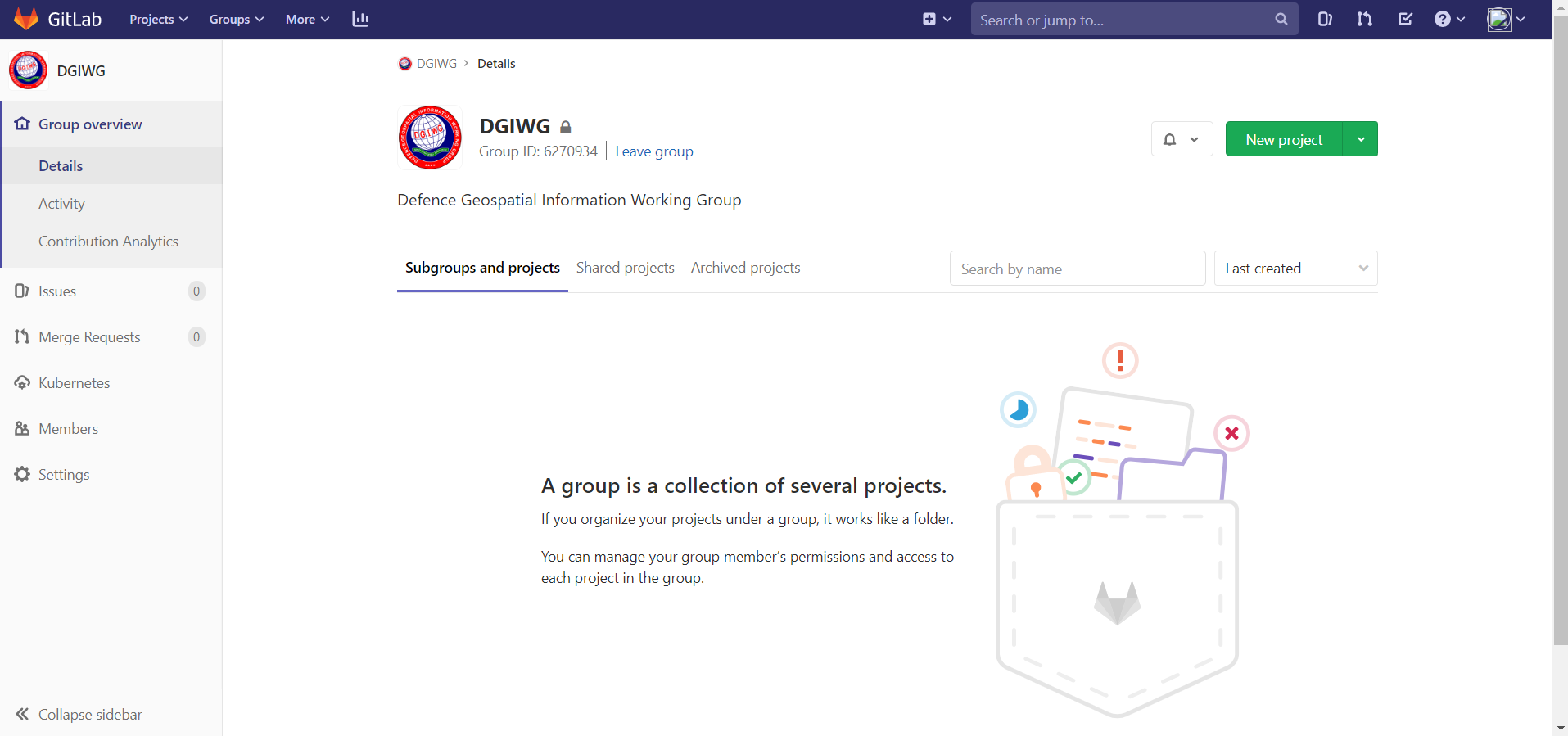
Draft documents have been generated to allow easier reading and commenting.



**Figure 5. DGIWG HTML rendering from the current draft template**

2.5. and what about GitLab ?

GitLab is another Git-based repository, offering more or less the same functionalities as GitHub. One of the main difference is that GitLab private repositories are free (public repositories are charged), whereas GitHub offers the opposite (public is free, private is charged). OGC has then investigated use of GitLab and now (announced the 14th of January 2020) offers its own flavor on <https://gitlab.ogc.org/>. OGC procedure is simple, "**Private Repos = GitLab / Public Repos = GitHub**". the process development is the same. DGIWG P5 has created the DGIWG group under GitLab, but its use has not been investigated yet.



**Figure 6. DGIWG Group created under GitLab**

3. Feedback

3.1. Advantages

1. installed tools allow easy offline editing, contributions before meetings
2. merging contributions is an easy step and is managed by github.com web site (pull requests). Differences (additions, deletions…​) are highlighted, discussions during meetings or teleconferences becomes easier.
3. managing references, tables and images is much easier (at least it is automatic based on a simple syntax and no error is done as hen exchanging a word document between people/different Office version). NOTE: the cleaning part of a DGIWG standard is a demanding task (which even with care leads to some mistakes, error references in some DGIWG published documents).
4. using same technologies as OGC may also be an advantage for DGIWG when profiling OGC standards (of interest for Web services and upcoming OGC APIs). This also can been as an advantage for nations aiming at developing national profile based upon DGGIWG standards.
5. becoming friendly with the tools and the DGIWG proposed workflow requires only a few hours.

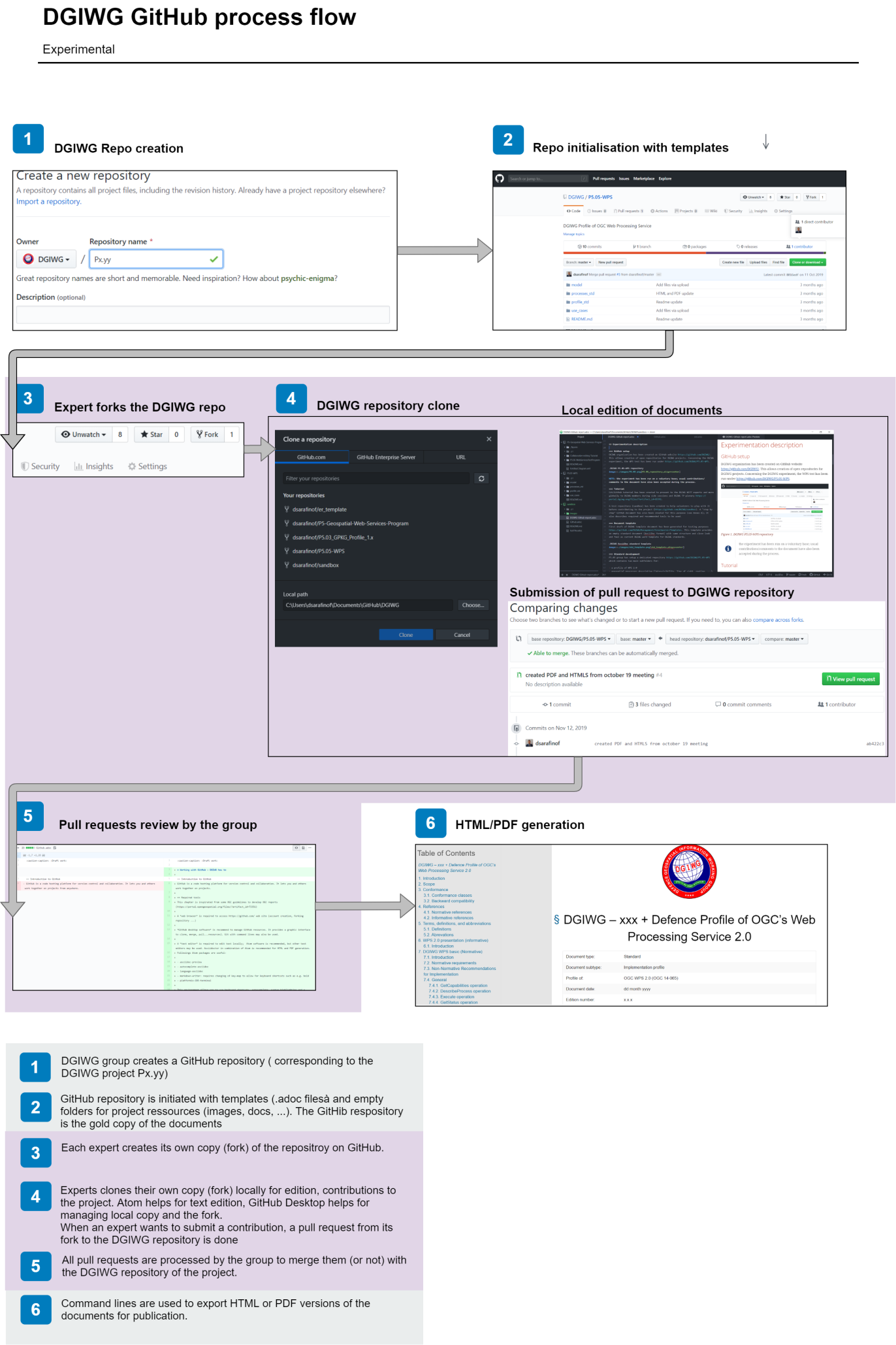
3.2. Things to be checked

1. Tools installation (see [[Annex]](file:///C:\Users\dsarafinof\Documents\GitHub\DGIWG\sandbox\DGIWG-Github-report.html#Annex))
2. Access to <https://github.com/> (only for specific tasks)

3.3. Recommendations and way forward

1. **Continue P5.05 experiment** and finalize it with GitHub tools
2. **Finalize DGIGW documents template in asciidoc** (+ CSS)
3. **Review all resources** to support using GitHub for DGIWG
4. **Extend experiment to other DGIWG panel(s)** (volunteers ?) to gather wider feedback
5. **Setup management rights** on <https://github.com/DGIWG/>
6. **Start investigation of GitLab** in line with OGC process

4. Annex A



**Figure 7. DGIWG simple flow for standard developpment under GitHub**

5. Annex B

Working with GitHub : DGIWG how to

6. Introduction to GitHub

GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects.

7. Required tools

This chapter is inspirated from some OGC guidelines to develop OGC reports (<https://portal.opengeospatial.org/files/?artifact_id=73351>)

A **web browser** is required to access <https://github.com/> web site (account creation, forking repository …​)

**GitHub desktop software** is recommend to manage GitHub resources. It provides a graphic interface to clone, merge, pull…​resources). Git with command lines may also be used.

A **text editor** is required to edit text locally; Atom software is recommended, but other text editors may be used. Asciidoctor in combination of Atom is recommended for HTML and PDF generation. Followings Atom packages are useful:

* asciidoc-preview
* autocomplete-asciidoc
* language-asciidoc
* markdown-writer: requires changing of key-map to allow for keyboard shortcuts such as e.g. bold
* platformio-IDE-terminal

This environment allows you to use keyboard shortcuts, autocomplete, syntax highlighting and a rendered preview for asciidoc; and provides you an terminal window within the editor to convert your asciidoc to html and pdf

8. Working on an existing repository

This first use case is useful when you want to collaborate to an existing project. That is the case for the DGIWG activity. In this case, the workflow is

* Go to the DGIWG project/repository you want to work on (for example <https://github.com/DGIWG/sandbox>)
* Fork the project to your account, so you have your own version.
* Clone on your machine your version of the repository (your fork that looks like <https://github.com/yourNames/sandbox>). This may be done with github desktop tool.

⇒ this will create a copy of your repository on you local hard drive.

* Edit file locally (for example with Atom editor tool, check "Recommended Asciidoc Environment" on <https://portal.opengeospatial.org/files/?artifact_id=73351> for installing the tool and related plugins)
* Save file locally (still with the text editor)
* Commit locally (for example with github desktop)
* Push to Fork (for example with github desktop)

⇒ this will push your local changes to your own branch of the repository (<https://github.com/yourName/sandbox>)

9. Pulling changes from your fork

* create a new pull request (from your own fork on github.com, for example <https://github.com/yourName/sandbox>)

⇒ this pull request has then to be discussed and accepted by admin of the DGIWG repo (<https://github.com/DGIWG/sandbox>)

* once your pull request has been accepted by the DGIWG repo, you just need to synchronize your fork again with it by (with Github desktop for example)
  + comparing your local copy to the DGIWG branch <https://github.com/DGIWG/sandbox>)
  + make a merge (this will update your local clone of your own repository), then your local Copy = the DGIWG online repo (<https://github.com/DGIWG/sandbox>)
  + push it to the origin (this will update your online copy <https://github.com/yourName/sandbox>), then <https://github.com/yourName/sandbox> = <https://github.com/DGIWG/sandbox>

10. Working on a new project

How to create a new project? The easiest is to use the Github UI to create a new project. You can also use the command line.

11. GitHub Flow

[Github Flow](http://1.bp.blogspot.com/-n8gwrM5Bf04/UfosDLuuDUI/AAAAAAAAKwg/2aE3V0NDk-g/s1600/git-and-github-workflow.png)

Last updated 2020-01-21 09:03:28 +0100