

Table of Contents

DGIWG GitHub report	2
1. Introduction	3
1.1. Context	3
1.2. Aim	3
2. Experimentation description	4
2.1. GitHub setup	4
2.2. Tutorial	4
2.3. Document template	4
2.4. Standard development	5
2.5. And what about GitLab ?	7
3. Feedback	8
3.1. Advantages	8
3.2. Somme identified issues	8
3.3. Things to be checked	9
3.4. Recommendations and way forward	9
4. Annex A	10
5. Annex B	12
Working with GitHub : DGIWG how to	13
6. Introduction to GitHub	14
7. Required tools	15
8. Working on an existing repository	16
9. Pulling changes from your fork	17
10. Working on a new project	18
11. GitHub Flow	19



DGIWG GitHub report

Document type:	Technical Report
Document date:	27 January 2020
Edition number:	0.1.1
Responsible Party:	DGIWG
Editor :	Dimitri Sarafinof
Audience:	DGIWG participants and associates
Abstract:	This document describes the GitHub experimentation ran by DGIWG WSTP about using GitHub tools within the DGIWG specific standardization process.
Copyright:	© Copyright DGIWG, some rights reserved - (CC) (By:) Attribution You are free: - to copy, distribute, display, and perform/execute the work - to make derivative works - to make commercial use of the work Under the following conditions: - (By:) Attribution. You must give the original author (DGIWG) credit. - For any reuse or distribution, you must make clear to others the license terms of this work. Any of these conditions can be waived if you get permission from the copyright holder DGIWG. Your fair use and other rights are in no way affected by the above. This is a human-readable summary of the Legal Code (the full license is available from Creative Commons < http://creativecommons.org/licenses/by/2.0/ >).

Chapter 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 (with recommended tools).

PS : the NGA also develops its own profiles on Github (<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.

Chapter 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. For the DGIWG experiment, the DGIWG WPS profile development has been run under <https://github.com/DGIWG/P5.05-WPS>.

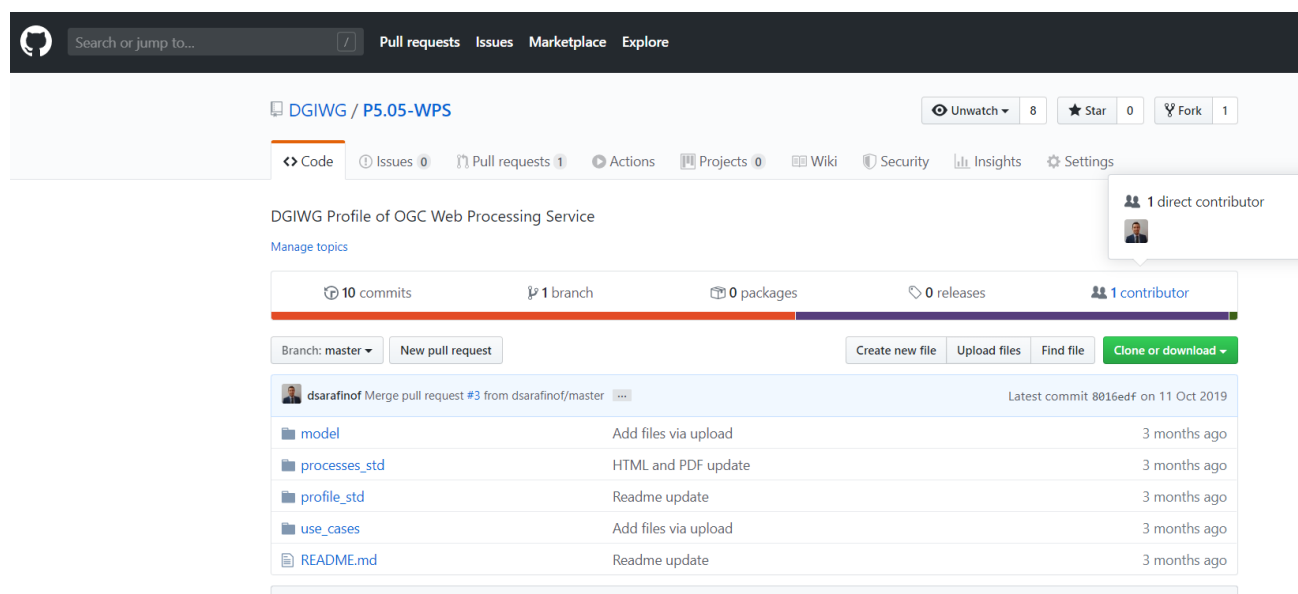


Figure 1. DGIWG P5.05-WPS repository

NOTE

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

2.2. Tutorial

A Git/GitHub tutorial has been created to be presented 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 (such as Atom text editor, useful plugins, AsciiDoctor ...)

2.3. Document template

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

current DGIWG word template for DGIWG standards.

DGIWG / Management

Unwatch

7

Star

0

Fork

0

Code

Issues 0

Pull requests 0

Actions

Projects 0

Wiki

Security

Insights

Settings

Branch: master

Management / Templates / Template_STD-TCR-HBKv0.2 /

Create new file

Upload files

Find file

History

dsarafinof

 update v0.2 template + html + pdf ouputs

Latest commit 4d4dd71 on 3 Aug 2018

..		
images	Template v0.2 for STD, TCR and HBK added	last year
resources	Template v0.2 for STD, TCR and HBK added	last year
00_Front_Material.adoc	update v0.2 template + html + pdf ouputs	last year
00_Introduction.adoc	Template v0.2 for STD, TCR and HBK added	last year
01_Scope.adoc	Template v0.2 for STD, TCR and HBK added	last year
02_Conformance.adoc	Template v0.2 for STD, TCR and HBK added	last year
03_Normative_Reference.adoc	Template v0.2 for STD, TCR and HBK added	last year
04_Terms_and_Definitions.adoc	Template v0.2 for STD, TCR and HBK added	last year
05_Subject_Matter_Content.adoc	Template v0.2 for STD, TCR and HBK added	last year
06_Subject_Matter_Content.adoc	Template v0.2 for STD, TCR and HBK added	last year
annex-a.adoc	Template v0.2 for STD, TCR and HBK added	last year
annex-b.adoc	Template v0.2 for STD, TCR and HBK added	last year
annex-bibliography.adoc	Template v0.2 for STD, TCR and HBK added	last year
annex-c.adoc	Template v0.2 for STD, TCR and HBK added	last year
annex-history.adoc	Template v0.2 for STD, TCR and HBK added	last year
index.adoc	Template v0.2 for STD, TCR and HBK added	last year
std.adoc	update v0.2 template + html + pdf ouputs	last year
std.html	update v0.2 template + html + pdf ouputs	last year
std.pdf	update v0.2 template + html + pdf ouputs	last year

Figure 2. DGIWG AsciiDoc standard template

2.4. Standard development

The 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, ...).

Contributions (with text editor Atom) have been made on each expert clone of the repository before submitting pull request to the DGIWG repository. See workflow and tutorial in the [Annex A](#).

NOTE

Atom is a text editor that let experts edit the text locally on their computer (see figure below)

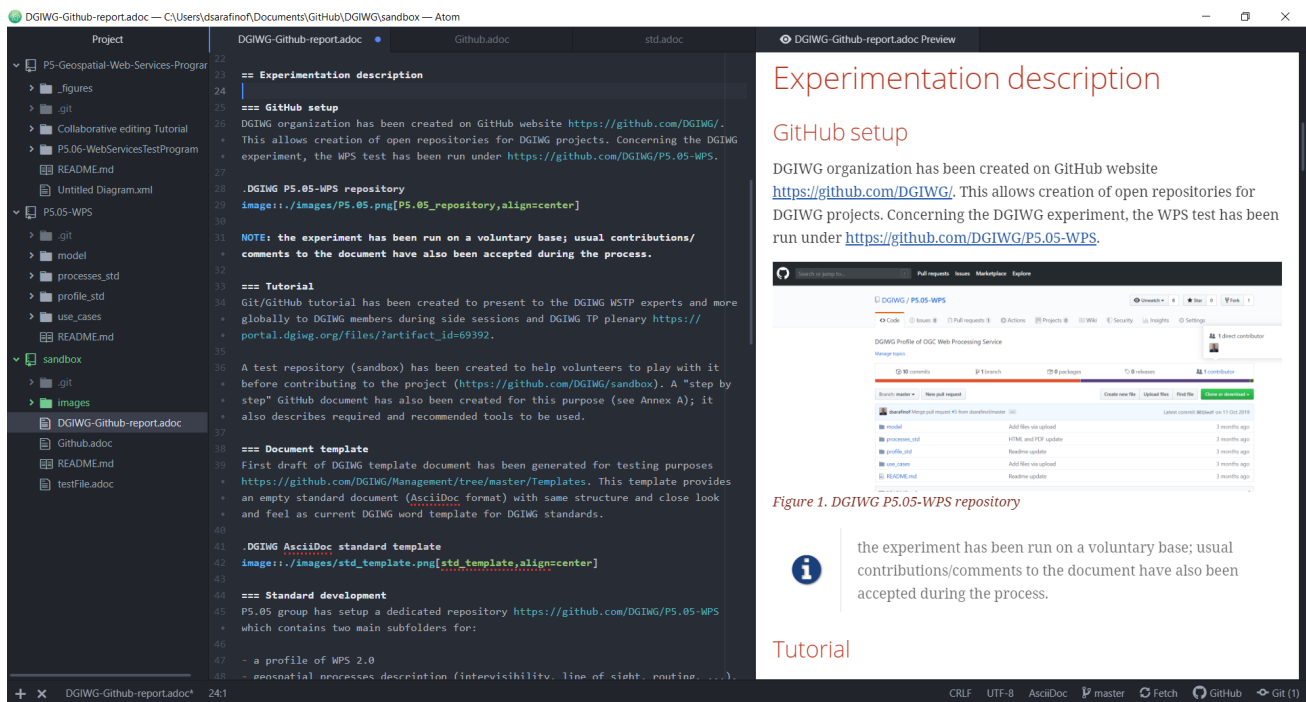


Figure 3. Edition of documents with Atom text editor and document preview

Local editions are submitted by experts on the DGIWG GitHub (pull request mechanism); differences are highlighted and easy to discuss (see figure below).

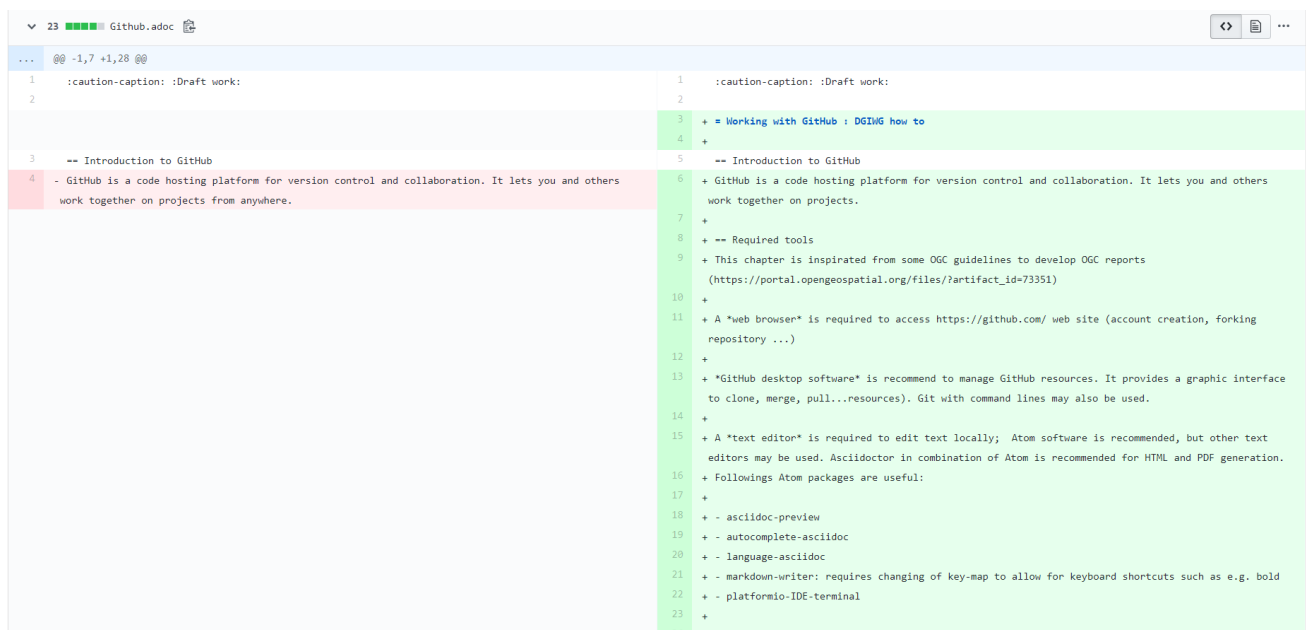


Figure 4. GitHub comparison during the pull request process

Each of these two documents follows the previous defined template. Command lines allow easy generation of PDF or HTML documents (draft documents have been generated to allow easier reading and commenting). This also can be done with a plugin of the Atom editor.

```
asciidoctor -r asciidoctor-bibtex -a linkcss -a stylesheet=dgiwg.css -a
stylesdir=resources/stylesheets std.adoc
```

Table of Contents

DGIWG – xxx + Defence Profile of OGC's Web Processing Service 2.0

1. Introduction
2. Scope
3. Conformance
 - 3.1. Conformance classes
 - 3.2. Backward compatibility
4. References
 - 4.1. Normative references
 - 4.2. Informative references
5. Terms, definitions, and abbreviations
 - 5.1. Definitions
 - 5.2. Abbreviations
6. WPS 2.0 presentation (informative)
 - 6.1. Introduction
7. DGIWG WPS basic (Normative)
 - 7.1. Introduction
 - 7.2. Normative requirements
 - 7.3. Non-Normative Recommendations for Implementation
 - 7.4. General
 - 7.4.1. GetCapabilities operation
 - 7.4.2. DescribeProcess operation
 - 7.4.3. Execute operation
 - 7.4.4. GetStatus operation



§ DGIWG – xxx + Defence Profile of OGC's Web Processing Service 2.0

Document type:	Standard
Document subtype:	Implementation profile
Profile of:	OGC WPS 2.0 (OGC 14-065)
Document date:	dd month yyyy
Edition number:	x.x.x

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 differences 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 the 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 also created the DGIWG organization under GitLab, but has not investigated its usability yet.

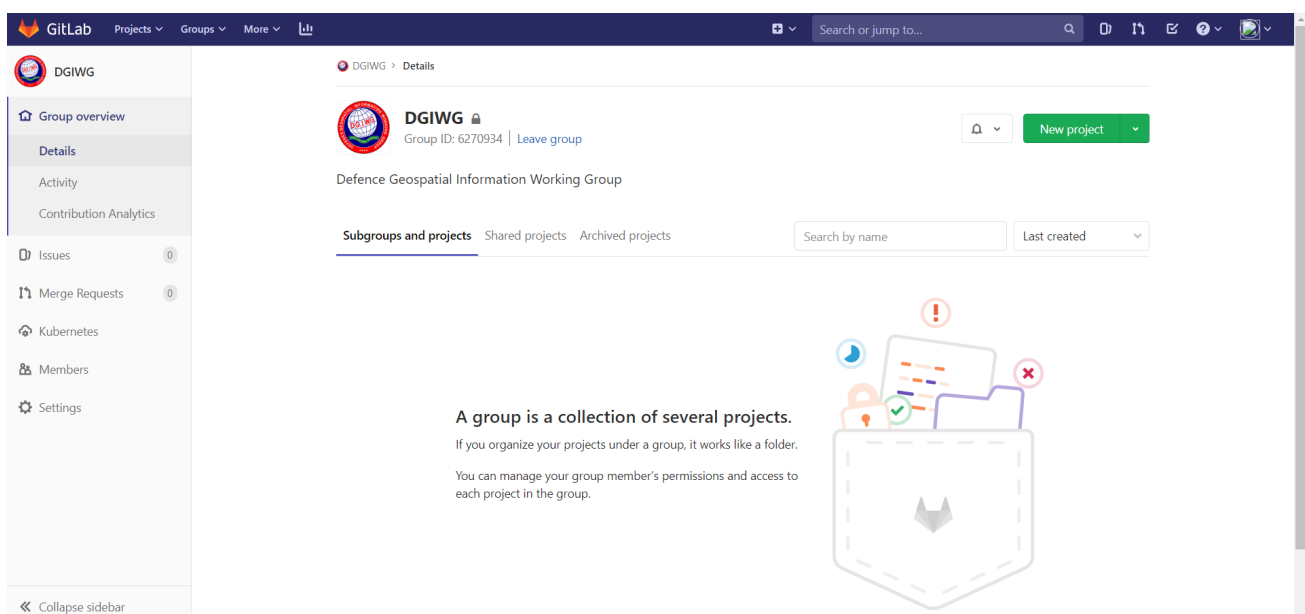


Figure 6. DGIWG Group created under GitLab

Chapter 3. Feedback

3.1. Advantages

1. publicly available and free tools allow easy offline editing and contributions between 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 introduced by exchanging a word document between people which use different Office versions or different language settings). NOTE: the final editing part of a DGIWG standard for publication as Word or PDF document is a demanding task (that even with most carefulness may lead to mistakes or erroneous references in published DGIWG documents).
4. using the same technology and workflow as OGC may also be an advantage for DGIWG when profiling OGC standards. This is especially of interest for Web services and upcoming OGC APIs profiles, but can also be an advantage for nations aiming at developing national profiles based upon DGIWG standards.
5. becoming friendly with the tools and the DGIWG proposed workflow requires only a few hours.

3.2. Somme identified issues

The proposal to use GitHub and contribute to a collaborative approach to developing DGIWG profiles and Standards, it is assumed that:

1. the user has access to the Open Internet both to GitHub and GitHub download App plugin sites from a Military Network.
2. the Network the user is working on does not have any JavaScript (transactional) restrictions.
3. the User has Administrative Rights to install on their respective computer.
4. the User does not have any restrictions of items such as the 'Command Line Interface'.
5. the Military network can interact with the GitHub/Lab and Local Machine Repo.

NOTE

These issues have been identified by GBR. Full analysis can be found in the document ["Using GitHub on Defence Military Networks"](https://portal.dgiwg.org/files/?artifact_id=70084) [https://portal.dgiwg.org/files/?artifact_id=70084].

3.3. Things to be checked

1. Tools installation (see [Annex B](#))
2. Access to <https://github.com/> (only for specific tasks)

3.4. Recommendations and way forward

1. **Continue P5.05 experiment** and finalize it with GitHub tools
2. **Finalize DGIWG 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

Chapter 4. Annex A

DGIWG GitHub process flow

Experimental

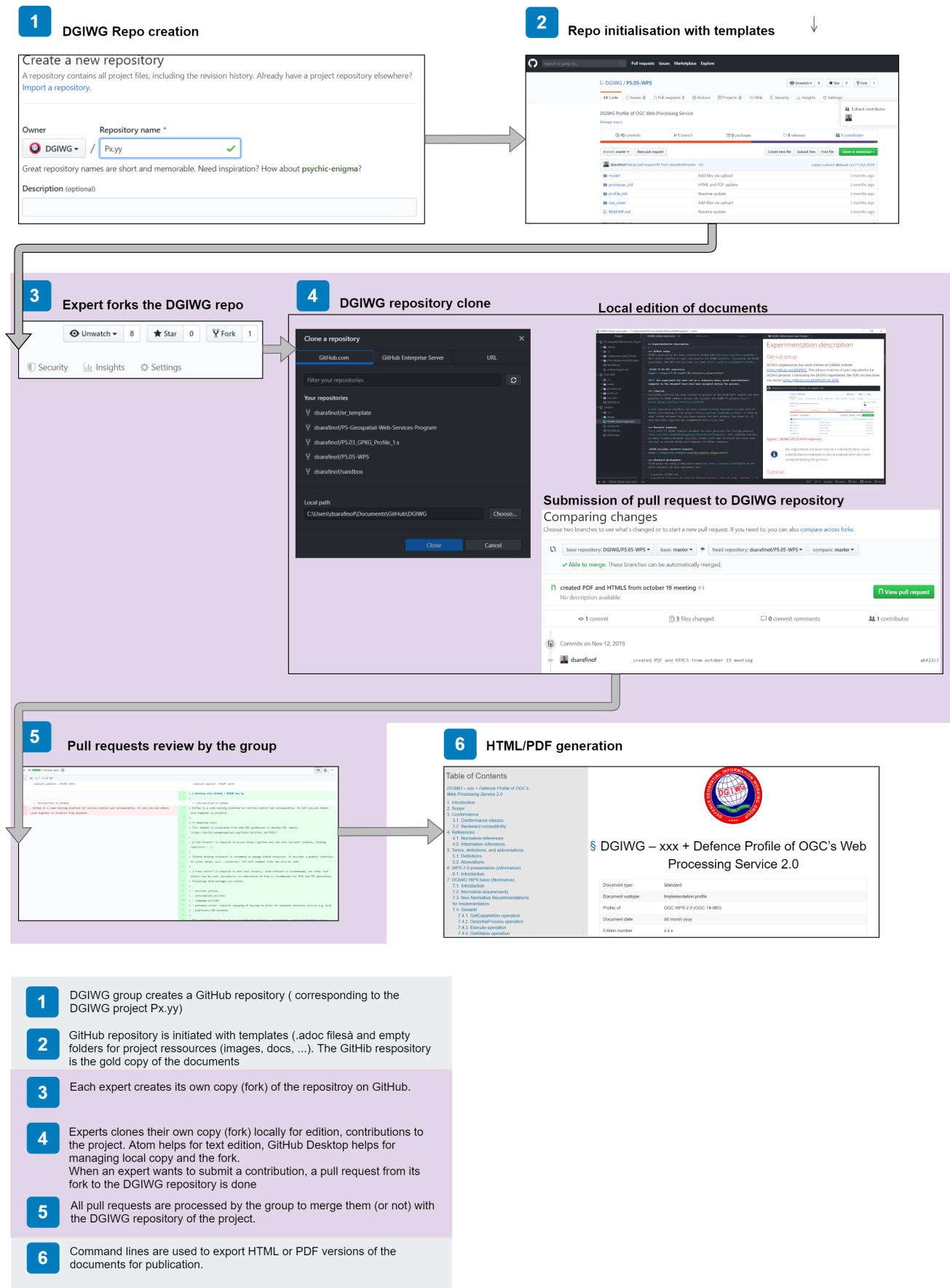


Figure 7. DGIWG simple flow for standard development under GitHub

Chapter 5. Annex B

Working with GitHub : DGIWG how to

Chapter 6. Introduction to GitHub

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

Chapter 7. Required tools

This chapter is inspired 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 ...)

Online editing

Online editing of asciidoc document is possible, directly with a web browser. This solution does not offer advance functionalities (visual preview, spellcheck...) that are available offline with dedicated tools.

Offline editing

GitHub desktop software is recommended to manage GitHub resources. It provides a graphic interface to clone, merge, pull...resources).

NOTE

Git command lines may also be used (without any graphic interface) to manage resources.

A **text editor** is required to edit text locally (offline); 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

Although GitHub can be used anonymously, a GitHub user account is required to interact with the platform and to be able to fork repositories. GitHub is free for public repositories.

Chapter 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](https://desktop.github.com/) [<https://desktop.GitHub.com/>].

⇒ 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 fork of the repository (<https://github.com/yourName/sandbox>)

Chapter 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>

Chapter 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 and then clone it. You can also use the command line (more info [here](https://help.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line) [https://help.github.com/en/github/importing-your-projects-to-github/adding-an-existing-project-to-github-using-the-command-line]).

Chapter 11. GitHub Flow

GitHub Flow [<http://1.bp.blogspot.com/-n8gwrM5Bf04/UfosDLuuDUI/AAAAAAAAAKwg/2aE3V0NDk-g/s1600/git-and-GitHub-workflow.png>]