Report on the implementation of a collaborative DMQC toolkit

Ref.: D2.5_V0.3

Date: 19/12/2020

Euro-Argo Research Infrastructure Sustainability and Enhancement Project (EA RISE Project) - 824131

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 824131.
Call INFRADEV-03-2018-2019: Individual support to ESFRI and other world-class research infrastructures

Disclaimer:

This Deliverable reflects only the author's views and the European Commission is not responsible for any use that may be made of the information contained therein.

Document Reference

Project	**Euro-Argo RISE - 824131**
Deliverable number	**D2.5**
Deliverable title	**A report on the implementation of a collaborative DMQC toolkit**
Description	**A report on the implementation of a collaborative DMQC toolkit (e.g codes and datasets repositories, organization of development and dissemination workflow, biennal training sessions)**
Work Package number	**2**
Work Package title	**Improvement of the core Argo mission**
Lead Institute	**Ifremer**
Lead authors	**G. Maze, K. Walicka, I. Benavides, G. Notarstefano**
Contributors	**C. Cabanes**
Submission date	**05/01/2021**
Due date	**[M24] December 31st, 2020**
Comments	
Accepted by	**G. Maze**

Document History

Version	Issue Date	Author	Comments
0.1	Dec. 10th, 2020	G. Maze	First draft to be circulated
0.2	Dec. 11th, 2020	K. Walicka	Edits on the DMQC survey components
0.3	Dec. 19th, 2020	G. Maze	Final version sent to project coordination

EXECUTIVE SUMMARY

Within WP2, our goal is to improve the quality of the Argo dataset. Beyond improving quality control methods, Task 2.4 assumes that an improved collaboration between Argo partners is the best way toward a long term and robust development of Argo (DM)QC activities, which ultimately lead to an improved Argo dataset.

Using a thorough assessment of the Argo community historical practices with regard to QC activities, during a WP2 meeting held in Dec. 2019, we developed the Euro-Argo collaborative framework strategy to achieve the objective of improving Argo (DM)QC activities.

This strategy encompasses 3 domains:

- Software (eg: development, performance, usage, access)
- Reference dataset (eg: content, access, availability)
- Data & expertise (eg: training, sharing, educating users)

To implement this strategy, we have set-up in Dec. 2019, and developed throughout 2020, an online collaborative toolkit at github.com/euroargodev. This deliverable provides a complete report on the collaborative framework strategy and its implementation.

The toolkit is organised around “repositories”: a collection of files (possibly) with online collaborative services like discussion threads (“Issues”), wiki pages, project management boards and a complete set of community development tools (based on “git”). The content of each repository can be shared through web sites or published with packages (eg “npm”, “docker”, etc …) or a simple zip file release. The Euro-Argo partners have agreed to use this toolkit as the primary tool to share, distribute and work together on Argo community tools. This has been started during this first year. More than 45 people have registered and more than 20 repositories are being filled with Argo useful information and tools.

TABLE OF CONTENT

Toward the Euro-Argo Collaborative Framework **7**
Rationale 7
Elaboration 7
Strategy 8
Software 8
Reference database 9
Data & expertise 9
Proposition 10
Description of the Euro-Argo collaborative framework **10**
General presentation 10
Membership and Teams 11
Tools and usage examples 12
Collaborative software development 12
Forums (“issues” and “discussions”) 13
Wiki 14
Website hosting 14
Project Management 15
One year achievements and assessment **15**
Quantitative aspects 16
Repositories and softwares 16
Existing softwares uploaded and now distributed by EA-CF 16
New softwares created and developed on EA-CF 16
Collaborators and Teams 18
Argo QC forum 18
Other key components of the EA-CF 19
Reference database 19
Optimised access 19
Improve content 19
Data & Expertise 20
Cookbook 20
Report templates 21
DMQC workshops 21

Future work **21**
About the EA-CF 21
Softwares 22
Reference database 22
Data & Expertise 22
Tables **23**
Figures **25**

Toward the Euro-Argo Collaborative Framework

Rationale

The objective of task T2.4 is to improve the Delayed Mode Quality Control (QC) process at the European level, with the ultimate goal to improve the Argo dataset overall quality.
To improve Delayed Mode QC and more generally all Argo QC activities is not easy. There are many reasons for that, among which one can highlight the following[1]:

- QC tasks involve a very wide set of technical and scientific skills that are difficult to build, from Argo data processing chain to regional oceanographic expertise,
- QC activities are distributed among different international groups and institutes, so that a lot of coordination is required,
- QC performances depend on reference databases that could be incomplete,
- Argo floats and sensors technology are continuously evolving, which increases the variety of technical parameters to manage (with an asymptotic horizon set by the lifetime of a specific technology, which is about a few years),
- QC expertise is preferentially "encoded" into operator skills rather than QC tests design, which complicates training,
- QC in marginal seas is an expertise required by Euro-Argo (EA) partners that is very specific and still lacks the long history of the standard core Argo QC,
- Softwares used for QC are mostly based on licensed programming languages (eg: Matlab), which impedeseasy training and sharing among international partners and non-academic groups,
- Most QC softwares are developed individually by local groups, which limits their long term availability and sustainability
- There is no Argo QC training material available, which, combined with very rare QC workshops strongly limit the size of the QC community,

The originality of WP2 and T2.4 is that all partners (IFREMER, IEO, FMI, OGS, NOC, BSH, SOCIB) decided to continue and insert their QC activities within an entirely new and innovative framework: the **Euro-Argo Collaborative Framework (EA-CF)**. The goal of this initiative is to tackle challenges and address difficulties listed above.

Elaboration

The EA-CF has been discussed all along the first year of the Euro-Argo RISE project (2019) and elaborated during the first WP2 intermediate meeting (Nov. 19-20th, 2019, Brest). During this meeting, we reviewed results from the DMQC survey conducted by NOC/BODC in 2019 and the status of existing QC softwares, tools, training material and reference databases.

BODC compiled a review of the current state of delayed-mode quality control tools and methods used within the global Argo community (DMQC survey report). According to the reports of DMQC operators, this document highlights the main barriers encountered while processing core Argo parameters in delayed mode, as well as the opportunities to enhance the sustainability and efficiency of Argo dataset and DMQC procedures. The key results of this survey are:

[1] as of late 2018, at the time of EARISE proposal writing.

- It was found that organisations reported a limited number of DMQC operators each, with a core community of experts reaching about 40 people worldwide. This points to the crucial need for DMQC training in order to enlarge the community.
- The vast majority of softwares used for DMQC are encoded in Matlab, but the community complains about the price of licences and the difficulty of working across very different Matlab versions that could ultimately lead to differences in QC diagnostics results. Obviously, this points toward the progressive development and adoption of softwares based on free languages, like Python.
- It was also found that a broad range of CTD and Argo reference data (ref-db) versions were used and that the content of the ref-db was often complemented by other independent/regional database. This points to the need for easy update methods and access to the last version of the ref-db and for improved content.
- Lastly[2], the lack of QC report sharing pointed toward the need for a community driven QC report template.

As a result, the survey allowed to identify the major needs to improve Argo QC activities:

- Routinely DMQC trainings
- Free open-source programming language for DMQC analysis
- An uniform DMQC GUI and other common tools for the entire Argo community
- Improvements in reference database (for shallow seas, data selection and metadata)
- Sharing repository with information about problematic floats (list and reports)
- Consider the other opportunities for improvements in methods to DMQC process

Based on these results, all partners discussed what should be the scope of the EA-CF by answering questions such as:

- Target audience and user base (Euro-Argo, national, international)
- What do we want to share? (code, data, etc…)
- What do we want to develop together, or not?
- Where is collaboration required to improve QC? (eg: code, data, expertise, …)
- Available collaborative tools (eg: code repo, online software, forum/issue/wiki, …)
- Where are performance bottlenecks? (eg: data selection, access, colocalization, ...)

Participants split into several working groups dedicated to discuss requirements and services to be provided by the EA-CF. Key points outlined by these working groups were presented and all participants endorsed the EA-CF strategy during a plenary session of the meeting.

Strategy

The strategy at the core of the EA-CF encompasses 3 domains.

Software

- About the **design**: promote modularity and Application Programming Interface (API) usage
- Encourage **co-development**:
  - Use code versioning (eg: git)
  - Organised codes in modern repositories (see eg: github.com/ArgoDMQC, github.com/ArgoRTQC)
- Pay attention to **performances**:

[2] We encourage the reader to explore the exhaustive survey report for more.

  - Improve performances with regard to Argo related data read access from disk but also cloud (eg: parallelisation, lazy data model)
  - Investigate how to optimise statistic methods performances (e.g. inversion, colocalization, interpolation)
- Provide **documentation**: it is crucial to produce for each software a complete usage and api documentation, but also typical Argo use cases, and possibly demonstration videos
- Should be **portable**, i.e. platforms agnostic (should run on Windows, Linux, Mac)
- Think twice about the choice of programming **language**: promote open-source and free to use languages (eg: python, R). Matlab is to be avoided for new softwares.

Reference database

- Improve ref-db **content**:
  - It is necessary to revise the selection criteria for a specific usage, typically for marginal seas
  - It is also necessary to gather all possible CTD casts among partners to increase the content of ref-db used in marginal seas
- Optimisation of ref-db **access**:
  - A single entry point used by all QC softwares should be developed (eg: erddap)
  - Storage format should be revisited: historical format is mat files, but other formats could help optimising read access (eg: nc, sql, zarr, gcs)
  - Work on performances to select and retrieve as fast as possible (meta-data vs data) ref-db content. Since this is one key bottleneck of the QC process, an API design should be produced to improve such performances
  - Ensure that QC methods make use of the last version of ref-db (eg: through API fetching from online resources)
- Modernize **authentication** method for ref-db access:
  - Ref-db content may be available with limited access, current access method (by exchange of emails for registration and single login/password retrieval) is obsolete
  - Ref-db access with authentication should be managed within softwares (eg: API keys)

Data & expertise

- Fully support community **training**:
  - A DMQC training workshop where new operators and experts can meet should be organised regularly, for instance every 2 years (1st was in 2018, the 2nd planned for 2020 has been postponed to 2021 because of the global COVID19 crisis).
  - Online material should be made available and formatted for new operators to self-train (decreasing the steep QC learning curve).
  - All training material should be self-content and freely available online.
- **Sharing**:
  - Is about developing the material to be shared:
    - An exhaustive DMQC cookbook with contributions dedicated to EA related issues, like marginal seas specificities in QC
    - Use cases of QC softwares
    - DMQC reporting, like QC report templates
    - DMQC workshops presentations and tutorials
  - and diversifying methods of sharing beyond classic mailing-list:
    - Modern “issues” on repositories, similar to an online forum
    - Videos (webinars, screen captures),

- Online notebooks with reproducible examples in self-contained environnement (eg: Binder)

Proposition

From this 1st WP2 meeting, it was concluded that the EA-CF should provide all the necessary tools to improve activities and collaboration among partners with regard to the **3 strategic domains**:

1. Software (eg: allowing for co-development, performance and usage improvements)
2. Reference database (eg: content, access, availability)
3. Data & expertise (eg: sharing methods and production of content)

This tight collaborative and code-sharing strategy will allow for the DMQC process to be more sustainable, transparent, and easier to implement for groups managing modest fleets of Argo floats or simply freshly involved in the complex task of DMQC.

All partners agreed to make their new DMQC material compliant with the EA-CF strategy.

Taking into account all the above mentioned requirements, IFREMER proposed to use the Github platform to implement the Euro-Argo Collaborative Framework. This is a code hosting platform for version control and collaboration. It lets users and others work together on projects from anywhere. In practice, even though Github is primarily a platform for software development, its full set of services allow it to answer or facilitate some of the requirements listed above, as will be described in the next section.

Description of the Euro-Argo collaborative framework

General presentation

The EA-CF main entry is located at:

https://github.com/euroargodev

Technically, it is associated with the Github "organisation" named *euroargodev.* The ERIC Euro-Argo has obtained the non-profit organisation status, which makes this solution for the EA-CF completely free to EA and its members. Given the actual number of users (see below), this would otherwise cost about 4500€/year.
A non-exhaustive list of available features are:

- unlimited collaborators on unlimited public repositories with a full feature set,
- unlimited private repositories with a limited feature set
- GitHub Community Support
- Dependabot alerts
- Two-factor authentication enforcement
- Team discussions
- Team access controls for managing groups
- 2,000 GitHub Actions minutes
- 500MB GitHub Packages storage

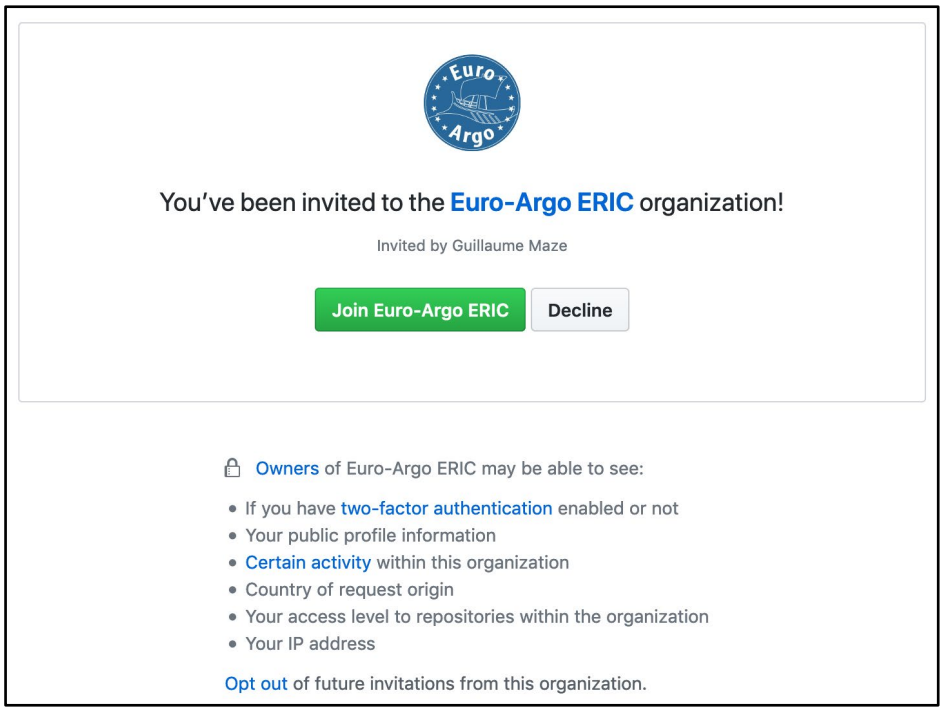
Other features will be described in the next section with real use case examples.

Figure 1 shows the EA-CF as it was on Dec. 9th 2020. In the upper part of the screen, one can see the Euro-Argo logo and general descriptive information about the organisation. Below this title/heading section are located tabs to access the different organisation main and useful sub-spaces:

- Repositories: this is the default landing page that provides the list and key information about all the repositories hosted under the EA organisation.
- Packages: this links to the list of possible packages distributed by EA. A software package is a service allowing to host software packages privately or publicly and to use packages as dependencies in our EA projects (eg: docker images and containers, npm, etc…).
- People: this links to the list of members, or outside collaborators, of the organisation.
- Teams: this links to the list of teams used to "organise" people participation in the organisation (see below).
- Projects: this links to the list of projects hosted at the organisation level (*Projects* are also a feature at the repository level).
- Settings: this links to a full set of settings concerning the organisation.

Membership and Teams

Anyone with a Github account (which can be created using an institutional email) can be enrolled as a member of the EA-CF. Enrollment is done by invitation only.



Message received by a Github user being invited to join the Euro-Argo ERIC organisation on the EA-CF at http://www.github.com/euroargodev.

Once a Github account has accepted the invitation, it is assigned a "role" within the organisation, which can be "Member" or "Owner". The key difference is that owners can manage all settings of the organisation, and more importantly are able to send invitations to become new members of the organisation.

During an initial 4 months set-up period, invitations were managed only by IFREMER. Then during the March 2020 Euro-Argo RISE General Assembly one "owner" was identified in each of the EA and Euro-Argo RISE participating institutions, to make enrollment easier and not too centralised. As of December 2020, the EA-CF participants with an "owner" role are:

- Ingrid M. Angel Benavides from BSH (Germany)
- Allan Berry from the Marine Institute (Ireland)
- Dimitris Kassis from HCMR (Greece)
- Guillaume Maze from IFREMER (France)
- Catherine Schmechtig from Sorbonne University (France)
- Simo-Matti Siiriä from FMI (Finland)
- Pedro Vélez from IEO (Spain)
- Kamila Walicka from BODC (United Kingdom)

We created teams for each of the institutions and partners of the ERIC Euro-Argo, as well as thematic teams, dedicated to people working on a specific project or thematic (eg: the team "core_argo_qc" gathers core Argo data QC experts). Teams can be nested to allow sub-groups to be formed.

Teams can be given specific roles on repositories and benefit from a dedicated service for discussions. Figure 2 illustrates this for the "dataref" team that gathers people working on the Argo Reference database. This service of the EA-CF allows for teams to entertain a private forum and to discuss relevant matters. As can be seen in the upper part of the webpage, teams can also manage and participate in specific repositories and projects.

Last, it can be noted that any Github account can be listed as an "outside collaborator". This is a useful feature in the case where someone outside the EA community would like to participate in specific discussions or repositories (eg: sub contractors, non EA members).

Tools and usage examples

Hereafter, we will provide a summary and the key elements relevant to the Euro-Argo RISE TaskWP2.4 collaborative framework objectives.

Remark: It is beyond the scope of this document to describe the full list of features of the EA-CF provided by Github. For this, please refer to the Github online documentation at:

- https://github.com/features
- https://docs.github.com/en

Collaborative software development

The EA-CF provides the most complete and modern set of features to develop softwares and collaborate on code development. This is due to the fact that Github is currently the leading platform for such features. Github has more than 40 millions users and its core goal is to facilitate version control and issue tracking for software development.

Code versions are managed with git, a free and open source distributed version control system. The purpose of git is to manage a collection of files and to track their changes over time. The local user folder holding files to be tracked is called a *repository*. When a user wants to take a snapshot of files, one creates a *commit*. A lot of information is saved with each commit about who, when, why (message) it was done. Git saves all the history of commits, and since each commit has a unique ID (hash), it is possible to restore (this is a *checkout*) files at different points in time. To backup all the history and content of a repository, one needs a safe non-local location, i.e. a remote place. The most popular remote is Github.

Collaborative development is then possible in 2 ways:

- Contributors use the same remote as a backup of their local developments (file changes). This can be used in small projects with few collaborators, since this is prone to conflicts in the versions of the files and lead to extra-work to reconcile all of them.
- Contributors create *branches* that are used to temporarily make changes or additions to the code. This is typically used when a contributor wants to fix a specific bug or add a new feature to the software. Once the new code is ready, the *branch* can be *merged* with the main code, which is typically located on the *master* branch.

Let's take examples to illustrate the EA-CF features for collaborative software development.

Figure 3 shows the landing page of the OWC salinity calibration software in python on the repository named argodmqc_owc. Useful key indicators for collaborative development are visible. These include: the list of direct contributors to the code (generated automatically), the explicit mention of last file updates, the commit messages of these updates, the last repository default branch update, message and author, and many more.

When a collaborator wants to contribute to a software, it will create a new *branch* from the original code and document its intention and work in a *pull request* (PR). This can be done to fix a bug or to create a new feature. Figure 4 illustrates this for the Argo data fetching software in python on the repository named argopy. Each PR can be "assigned" to a one or more collaborators and furthermore request for reviewers that will be in charge for the evaluation of the new code or code changes. Once reviewers have validated changes, one of the owners of the repository can merge the changes to the main code, closing the cycle of contribution.

Forums ("issues" and "discussions")

The EA-CF provides 2 different ways for collaborators to discuss and exchange on relevant topics.

Each repository hosts a service entitled "**issues**": this is a place where one can raise an issue about the software. Since this service works like a classic forum, once an issue is created, anyone connected with a Github account can post comments, populating the issue web page like a discussion thread. Like PRs, issues can be assigned to collaborators, labelled and linked to other services of the EA-CF (eg: projects, PR). Figure 5 shows an example of an issue raised by an user asking for more documentation about a software. The issue was assigned to the main developer. Once the issue is answered, addressed, it can be set to *closed* and will not be displayed on the default landing page. Issues are very useful to report bugs, un-expected behaviours or missing features. This is usually a starting place for a collaborator before creating a new development branch and a PR, typically addressing an issue (hence the possible link between PRs and issues).

EA-CF issues can contain formatted text but also code snippets and figures, making easy the documentation and illustration of a topic to discuss. Figure 6 illustrates these capabilities. Any type of document can be attached to an issue with a simple drag and drop (eg: pdf or excel sheets). If possible, the document content will be displayed online, otherwise a link will be created.
An additional feature associated with "issues" is the possibility for a logged in user to *subscribe* to an issue thread. Commenting or interacting[3] with an issue will automatically trigger the subscription of this user to notifications regarding this issue. But any user simply visiting and interested in staying informed about the issue being discussed can subscribe to be notified of any updates. Notifications are sent by emails, which provides a more classic way to receive information (eg: mailing lists). Notification settings can precisely be tuned on each user account settings page.

Further details on issues can be found here: https://guides.github.com/features/issues

The second tool that can be used as a forum on the EA-CF is located on the webpage associated with an organisation *team*. As was shown on Figure 2, a team webpage has a "**discussions**" tab allowing for a forum to take place between the members of a specific team. This can be very useful when a discussion is required on transverse topics. Moreover, team discussions are only visible to team members. Last, a team can be addressed from anywhere on EA-CF (i.e. in PR or issues threads) with the @ naming convention. For instance a BGC data issue raised on the public QC Forum can ask for an answer to the dedicated team by mentioning @euroargodev/bgc_argo_qc, this will trigger a notification to all members of the team.

Further details on teams discussions can be found here: https://docs.github.com/en/free-pro-team@latest/github/building-a-strong-community/about-team-discussions

Remark: At the time of writing this report, Github just released a new feature named "Discussions" that takes the teams discussion feature to the repository level. This new feature will be assessed in the coming weeks/months with regard to the EA-CF objectives.

Wiki

Repositories on the EA-CF can host wiki pages. These can be created, edited and managed directly online. This is a very simple way to provide documentation for a repository so that users can use and collaborate on a software. Examples of wiki pages can be found on the Argo QC Forum.

Further details on wiki pages can be found here:
https://docs.github.com/en/free-pro-team@latest/github/building-a-strong-community/about-wikis

Website hosting

Figure 7 gives several examples of websites built using the EA-CF. This is a more complex service but very powerful and useful to disseminate information. Basically, it is possible to publish as a website the content of a repository. Website url is built upon the organisation and repository names. For instance the website https://euroargodev.github.io/VirtualFleet is based on the repository content named *VirtualFleet*.

[3] users can react to an issue with icons.

It is possible to mix a software source code with the associated documentation on a single repository, as it is the case for the VirtualFleet software mentioned above (doc files visible on the website are located in a dedicated folder of the source code).
But it is also possible to build a repository from scratch, or using templates, with the only purpose of being accessed as a website (see the Argo online school and the Arvor-Provor workshop for examples).

Further details can be found here: https://pages.github.com

Project Management

To conclude this non-exhaustive overview of the EA-CF services, one can note that Github has Project Management capabilities as well. These were even considered at the beginning of the Euro-Argo RISE project before ActiveCollab was chosen. The interface for Project Management is centered on a *board* where cards in columns help users organize and prioritize work. Cards can have free text or be made of issues or pull requests. This is very handy to keep track and create a workflow to reach milestones and objectives.
Figure 8 shows an example of how this feature can be used to organize a project such as "Improve access to Argo data". Cards can be created automatically when issues are raised, and predefined actions allow to update cards position on columns. This can be used for instance to move a card from a column named "in progress" to "done" when it is associated with a PR or issue being closed.

Further details can be found here: https://github.com/features/project-management

One year achievements and assessment

The EA-CF was created in December 2019. This report thus comes at the end of a first year of existence that was dedicated to get to learn all the tools available and to train partners.
For the Euro-Argo RISE participants to get started with the Euro-Argo Collaborative Framework, IFREMER has created an online documentation that was presented during the 2nd EARISE General Assembly held in Paris in early March 2020 (see Figure 9). During half-a-day, IFREMER introduced the key elements and features of the EA-CF and animated a training session during which participants created their Github accounts and were invited to the *euroargodev* organisation.

Since participants identified collaboration on software development as the activity where they had the least expertise, emphasis was made on introducing the core concept and workflow of git and code versioning. Standard developers examples were adapted by IFREMER to the audience expertise in oceanography and Argo QC to make training material more appealing. The developed git introduction is available here: https://bit.ly/euroargodev_git.

The EA-CF and its key components (see Figure 10) was presented to the international Argo community during the last ADMT meeting (Dec. 3-4th 2020). The presentation is available here:
https://docs.google.com/presentation/d/1sZpWvHWdItfnz_loOTlpHcv0uWS-weXXaDNnxld5vqc

The EA-CF strategy and set of activities has received a warm welcome and encouragement from other Argo international partners. It was noted that this EA-CF initiative is one of its kind, that it is presumably the way forward to sustain Argo activities. This can be considered as an ADMT endorsement of the EA-CF.

Quantitative aspects

Repositories and softwares

As of Dec. 10th, 2020, there are 18 public repositories on the EA-CF. The full list is given Table 1. The primary languages used are Matlab and Python, but a total of 6 different languages is found.

Some repositories are being used primarily as forums, or online demonstrators:

1. **publicQCforum** for the Argo QC forum.
2. **recovery** is a forum for all matters concerning the recovery of Argo floats
3. **User-Acceptance-Test-Python-version-of-the-OWC-tool** a forum that is self explanatory, maintained by BODC.
4. **erddap_usecases**, was developed to host online examples showing how to access Argo data with Ifremer erddap. This repository will be deprecated in 2021 as content is moved to the Argo online school and argopy documentation.

Some repositories are used for websites production:

1. **argoonlineschool** is about the Argo Online School training and documentation website (Class 3) developed within Euro-Argo RISE WP7, maintained by IEO
2. **techworkshop** is about the Arvor-Provor technical workshop organised in January 2020
3. **euroargodev.github.io** is about the main entry page of the EA-CF (still in development)

The remaining repositories are associated with softwares. The EA-CF led to the online publication and centralisation of 9 softwares dedicated to Argo. These can be sorted in 2 categories.

Existing softwares uploaded and now distributed by EA-CF

These softwares existed before the EA-CF creation and/or use the EA-CF as a distribution medium. Indeed, partners agreed that a preliminary step toward a more intensive collaborative development of softwares would be to upload codes to EA-CF and to use the platform as a new distribution point. This was a great step toward centralizing information and making available a collection of softwares used in Argo QC activities.
These softwares are:

1. **check_CTD-RDB** dedicated to Checking the status of the CTD Reference database (RDB), maintained by BSH,
2. **dm_floats**, a set of Matlab tools for creating OWC input files and writing D_files from OWC results, maintained by INOGS
3. **matlab_profiles_visualization**, a Matlab GUI to work with Argo data, maintained by INOGS
4. **LOCODOX**, a Matlab Software for processing Oxygen data from Argo Float, maintained by Ifremer,
5. **DIVAA**, a Display Interface for Velocity And Argo data for the Argo-France program, maintained by Ifremer.

It is interesting to note that the Matlab language is used here on 4 out 5 softwares in this category.

New softwares created and developed on EA-CF

On the other hand, other softwares fully embraced the EA-CF features and are being actively developed collaboratively.
These softwares, that may use more than one repository, are:

- **pyowc**: an Argo float salinity calibration software in python, maintained by BODC. It is based on:
  - argodmqc_owc for the core software code
  - User-Acceptance-Test-Python-version-of-the-OWC-tool for interactions with beta testers
- **argopy**: a python library for Argo data beginners and experts, maintained by Ifremer. It is based on:
  - argopy for the core software code
  - argopy-data to host tutorials, examples and tests data
  - argopy-status to provide continuous monitoring of online resources used by argopy
- VirtualFleet, a numerical Argo fleet simulator developed as part of Euro-Argo RISE for Task2.3, maintained by Ifremer
- dm-report-template, a DMQC report templating software, maintained by BODC

Contrasting with the previous software category, one notes that the Python language is used on 3 out 4 softwares in this category.

Focus on **pyowc**: a new salinity calibration software in python

Developed by BODC with contributions from Ifremer, this new software is one key component of the EA strategy to address major challenges in DMQC activities (see the EA-CF rationale and elaboration section) such as using a free coding language (python), improving performances and promoting collaborative development and modularity of DMQC tasks (eg: data fetching, visualisation). The majority of software conversion has been done as a part of MOCCA project. The objective of the pyowc software developments were to:

- Retain all existing functionality of the Matlab version,
- Achieve the same outputs as the Matlab version,
- Enhance the computational efficiency,
- Ensure the Python version of the code is suitable for integration into a GUI interface.

The first 2 points were determined critical to re-insure the DMQC community that has been using the Matlab OWC software for more than 10 years and could have been excessively cautious about new software. Hence the critical role of the user acceptance tests phase. The Matlab OWC software conversion in Python, achieved a milestone in 2020 by moving to the user acceptance tests phase involving a small group of DMQC operators. Further user acceptance testing, including broader DMQC Argo community, is planned for early 2021.
The original code hosted on a BODC git server was migrated to the EA-CF and significant refactoring of the code took place in 2020. As of today, the pyowc is close to a first public release, expected in 2021. Figure 11 shows examples of typical graphics used to calibrate salinity data with the software.

Focus on **argopy**: a python library for Argo data beginners and experts

Developed by IFREMER with contributions from 5 EA external collaborators, this new software aims to ease Argo data access and manipulation for standard users as well as Argo experts. This initiative fulfils the EA-CF strategic points of software components modularity and improved performances. The overarching purpose of argopy is to be used as module in other softwares to fetch simply and efficiently all Argo data. This includes, for instance: ref-db data for QC softwares such as pyowc (described above) or the Canadian bgcArgoDMQC but also online tutorials such as the Argo online School or trajectory data for the VirtualFleet. The full documentation of the argopy software is available online here.

One key idea in the design of argopy is that in order to ease Argo data analysis for the vast majority of standard users, argopy needs to implement different levels of verbosity and data processing to hide or simply remove variables only meaningful to experts. Thus, users fetch Argo data by providing a space/time domain or a float WMO. No other information is required. This makes data fetching *easy*.
Another key idea in the design of argopy is that whatever the source of the data selected by users (the Ifremer erddap, argovis or a local copy of the GDAC ftp), data fetching is optimised under the hood with massive parallelisation. This makes data fetching *fast*.
The argopy software encountered a large and warm welcome in the research community (9193 downloads, 62 stars and 14 forks on Github, as of Dec. 10th, 2020). This indicates it fulfilled a community need. A lot of encouraging messages were also sent (see Table 2).
The argopy software is fully documented in a peer reviewed open-source journal in the article:
Maze and Balem, (2020). argopy: A Python library for Argo ocean data analysis. Journal of Open Source Software, 5(53), 2425, https://doi.org/10.21105/joss.02425

Collaborators and Teams

As of Dec. 10th 2020, there are 49 people registered as members of the *euroargodev* organisation. The 49 people are listed and organised within 25 Teams. We count 3 more people formally listed as outside collaborators, but this does not include non members who actively participated in repositories (about a dozen more).
There are 36 people actively participating in Euro-Argo RISE WP2 activities. Thus we succeeded in engaging with 13 more people than expected and, added to repository contributors, one can state that the EA-CF attracted a reasonably large user base.
However, the Teams Discussions feature is not really being used so far, with very few activities observed after the initial set-up of a Team.

Argo QC forum

The "issues" feature of a Github repository can be used as an online forum. This, in practice, is how we created a public **Argo QC forum** on the EA-CF.

As mentioned in the rationale section, building a knowledge database and easing connection with or access to expertise is a key ingredient to improve Argo DMQC activities. Figure 12 shows the Argo QC forum landing page and its components. The forum has seen many issues raised and discussed among the partners. Main topics were:

- FORTRAN code for RTQC of bio-optical variables
- DMQC report template for Core Argo parameters
- User Acceptance Test: Python version of the OWC tool
- fast salinity drift
- COVID19 impact on Argo QC activities: suggestions and/or advice
- < example of sensor failure> Kistler pressure sensor issue
- 2nd European Argo/7th International Argo Delayed-mode QC Workshop for CTD data
- Status of Machine Learning for Argo QC
- Software available for core QC

We cannot anticipate for the long term impact on Argo QC activities of such a forum. Since QC specific activities are a rather slow process (on the order of several weeks to months), it will take a certain amount of time to populate the forum and see it fully adopted as the main medium for discussions about QC (at this point, it is still in "competition" with the international argo-dm and argo-dm-dm mailing lists). It is too early to determine if the 2 tools will find some complementarity or if one will prevail. At this point, we note that about one issue was raised every month, except during the summer break. This looks like a reasonable rate of use, basically reflecting the frequency at which discussion topics emerge among the community.

Other key components of the EA-CF

This report naturally focuses on the new tools made available by the online EA-CF toolkit. But, as stated in the rationale section, the WP2 Task 2.4 overarching goal is to improve all aspects of Argo QC activities. So partners further worked on activities of the other EA-CF strategic families, trying to adopt the logic of the EA-CF strategy described earlier.

Reference database

The Argo reference database is a key element for QC. Partners worked on two aspects with that regard.

Optimised access

Ifremer has made available on its erddap server, the Argo float data component of the full Argo ref-db: http://www.ifremer.fr/erddap/tabledap/ArgoFloats-ref.html
Before that, the classic method to access the Argo ref-db was to download from ftp a large zip file consisting of a collection of Matlab binary files. This was initially made for the Matlab OWC salinity calibration software.
An erddap access provides several improvements:

- only data relevant for the QC of a given float can be downloaded (space/time selection criteria are available);
- the data accessed are always the last available version, avoiding to work with obsolete or limited data;
- Lastly, data can be fetched from an online data access form, but also from machine to machine.

Ifremer has developed examples and use cases to explain how to make use of this optimised Reference dataset access point. These are available and can be run directly online on this EA-CF repository: euroargodev/erddap_usecases.
Ifremer furthermore implemented in the argopy data fetching software a simple option to access reference data from the erddap. An example of usage can be seen here and it was also documented in the DMQC cookbook (see below).

Improve content

BSH, in collaboration with FMI, IOPAN, INOGS, SOCIB and Ifremer have worked on improving the content of the Argo ref-db in the Baltic, Arctic/Nordic and Mediterranean Seas by developing specific

selection criteria of historical data (standard criteria are derived for the open ocean and not adapted to marginal seas).
These criteria allow DMQC operators in marginal seas to perform a custom selection of the reference profiles according to:

- the specific trajectory of each float (from both reference databases), for instance floats drifting in and out of Boundary Currents
- maximum recorded depth,
- additional quality flags. Eg. a suspicious profile position or Maximum Recorded Depth (MRD) larger than the Bottom depth

Some comments on Med Sea (and also Black Sea) reference dataset: shallower profiles (< 900 m) are now taken into account also by Coriolis. In general, we collect data from Med research institutes and try to gather data also by partners that usually don't share them because they are not involved in international project or regional data collection systems. Another source of data is the Copernicus Marine Environment Monitoring System (CMEMS). Collected data are then compared to the Coriolis CTD reference dataset and checked to remove duplicates. CTD data are eventually merged into the Coriolis CTD reference dataset and grouped according to the dimension of various climatological sub-basins.

A much more detailed description of all DMQC activities improvement in marginal seas will be provided in the upcoming D2.7 Euro-Argo RISE deliverable entitled "A report on the adaptation of existing DMQC methods to Marginal seas (Arctic, Baltic and Mediterranean)".

Standard Argo ref-db boxes have also been improved for a finer and more up to date description of the Nordic Seas. This resulted in a significant increase of the number of profiles in the ref-db, from 9460 to 14354 in the Nordic Seas (see Figure 13). A by-product of this CTD ref-db update (2019v1) is the repository **check_CTD-RDB** which allows the ref-db users to easily identify temporal and regional gaps in the available CTD reference data, which may have important implications for the DMQC results. In the future, DMQC operators will be encouraged to use this tool to perform a local audit of the ref-db for their region of interest, provide feedback for improvement and contribute with local updates.

Moreover, the 2019v1 CTD Argo ref-db update for the Nordic Seas, included the removal of out of range, incomplete samples and profile duplicates. For the upcoming CTD Argo ref-db update, to be released in early 2021, the same cleaning procedure was applied to the Weddell and Caribbean Seas, the North and South Atlantic basins, as well as the Baffin Bay. This version will also include further updates for the Arctic (between 20°E - 180°E) and Nordic Seas.

Data & Expertise

Cookbook

After the 1st Euro-Argo DMQC workshop in 2018, it was decided to create an online cookbook with all the preparatory work of the workshop. Developed by experts of the workshop, it contains in practice information to perform DMQC and to make decisions. It has 3 sections:

- General information for DMQC analysis
- Specific information for regional analysis
- Case studies. One or more case studies are described for 5 regions of the world ocean.

"Case studies" is a very new type of DMQC training material. They provide illustration of complex CTD issues and provide guidelines to fix or flag appropriately Argo measurements. Such issues are for instance the recent anomalously fast salty drift of SeaBird SBE41.
The cookbook also describes how to fill netcdf files with DMQC results and outcomes, a very useful section for new operators.

The full DMQC cookbook is available here[4]: bit.ly/DMQCcookbook

In 2020, as part of the EA-CF, the first draft has been finished and is now circulating within a larger audience at the international level and was presented at the ADMT21 meeting in 2020 (Figure 10).

Report templates

Documenting how DMQC decisions are made by experts is one key element to improve the overall reliability of DMQC activities, which, we recall, is distributed among different groups over the world. Toward this decision making documentation (and hence sharing), to adopt a single format for reports is a requirement. BODC is organising the creation of such a DMQC report template. A first draft is already available at https://github.com/euroargodev/dm-report-template.

DMQC workshops

Because of the COVID19 international crisis in 2020, the 2nd Euro-Argo DMQC workshop that was scheduled in May in Liverpool, UK has been cancelled. Some items were still discussed online, like user acceptance tests of the pyowc software, but no real workshop took place. It was decided to postpone the workshop to 2021 (see below).

Future work

There is still a lot to achieve to fulfil the EA-CF strategy among European partners, and possibly beyond. In the near future (2021), we will focus on the following elements.

About the EA-CF

The EA-CF is still very young, and fully adopting its usage will take time because it requires people to change deeply grounded habits. Explaining and showing by examples the benefits of the EA-CF will be required for the coming years.

As the amount of material and content will increase over time, it may become difficult to find access to the key component of the EA-CF. We therefore plan on organising content according to the EA-CF strategy and to make this visible and available on a website at https://euroargodev.github.io, (content will be managed on this repository https://github.com/euroargodev/euroargodev.github.io). This will become the front page of the EA-CF.

We will continue to assess how people use Github features and to train beginners, so that as many Euro-Argo members as possible (existing and future members) can benefit from these features to improve their Argo related activities.

[4] Original url: https://docs.google.com/document/d/18ijKY1SoLE-L3WmBM4kgUoAQEsQkNltasbnMUmfdmig

Softwares

In 2021 and beyond, we will need to work on closing the gap between community development tools and existing Matlab softwares of the EA-CF. We have seen that EA-CF Matlab-based softwares mostly inherit from pre-existing softwares and individual development environments. Migrating to EA-CF offers many possibilities that could be ceased to improve the quality of these softwares.

In the meantime, migrating to open-source languages will be promoted. Such a move has been correlated to a more sustainable and community driven path of software development. We will therefore need to keep training members and showcasing advantages with and of open-source languages such as python.

argopy is the only software that was able to attract a large user base and some visibility out of the EA partnership. This is due to its purpose (i.e. to fetch and manipulate Argo data easily, for beginners and experts) that corresponds to a much larger user base than the Argo QC community targeted by the other softwares on the EA-CF. That being said, promoting EA-CF softwares to increase engagement from the community into our softwares use and development, is one key aspect that will need to be developed.

Reference database

With regard to ref-db improved **access**, our next priority is to implement a method to fetch the non-public ref-dev component (CTD casts from R/V cruises). With regard to the ref-db **content**: In Marginal seas, more criteria are still being worked on (Ex. Front side, Season/Month of observation, enclosed subbasin, etc).

Data & Expertise

Cookbook: after the recent presentation at the ADMT21 meeting, comments will be taken into account and possible updates made according to decisions of the ADMT (eg: on new recommendations for floats affected by the Fast Salinity Drift issue). The first version will then be released in early 2021. Since this is a living document, it will be augmented with new case studies and possible new QC issues.

DMQC workshop: During the 2nd WP2 intermediate meeting of Nov. 18-19th, 2020, it was decided to organise the 2nd DMQC workshop in June or July 2021, as a 100% online event. It is planned to consider feedback from the 1st edition post-meeting survey to address beginners' expectations. It is also planned to adopt a “modular” approach with the meeting organisation:

- Phase 1: “offline Argo QC school”, with training material that registered participants could access to and use to self-train independently (eg: cookbook, webinars, short tutorial videos and notebooks)
- Phase 2: “hands-on” online sessions (about 2h00 each) where registered participants run QC analysis with the help of experts
- Phase 3: “workshop” online sessions (about 2h00 each) where experts address current questions, issues and methodologies

Phase 1 and 2 are based on reusable content that will need to be prepared in advance. Phase 3 is a more standard experts meeting, exceptionally held online.

Tables

Table 1. List of EA-CF repositories, sorted alphabetically.

Name	Language	Created at	Pushed at	Issues	Forks	Stars	Size
argodmqc_owc	HTML	2 Jun 2020	17 Nov 2020	15	2	6	135123
argoonlineschool	Jupyter Notebook	30 Jun 2020	6 Aug 2020	0	1	1	44024
argopy	Python	17 Mar 2020	10 Dec 2020	14	14	62	3499
argopy-data		30 Mar 2020	28 Sep 2020	0	0	0	18076
argopy-status	Python	25 Sep 2020	10 Dec 2020	0	0	0	2331
check_CTD-RDB	MATLAB	2 Apr 2020	29 Oct 2020	0	0	2	45
dm-report-template	MATLAB	24 Nov 2020	24 Nov 2020	0	0	0	7631
dm_floats	MATLAB	29 Jun 2018	10 Jun 2020	0	0	0	133
erddap_usecases	Shell	27 Nov 2019	6 Apr 2020	0	0	4	19860
euroargodev.github.io		11 Feb 2020	11 Feb 2020	1	0	0	2
LOCODOX	MATLAB	19 May 2020	9 Nov 2020	0	0	0	107981
matlab_profiles_visualization	MATLAB	27 Mar 2020	7 Apr 2020	1	0	0	4411
publicQCforum		20 Nov 2019	4 Mar 2020	7	0	8	371
recovery		30 Jan 2020	31 Jan 2020	3	0	3	8
techworkshop	CSS	30 Jan 2020	15 Apr 2020	0	0	2	249055
User-Acceptance-Test-Python-version-of-the-OWC-tool		11 May 2020	11 May 2020	9	0	1	0
VirtualFleet	Python	26 Feb 2020	9 Dec 2020	6	1	0	7832

Table 2. Testimony about the argopy software developed as part of the EA-CF

"I've been playing around with argopy for the past couple weeks and have been very impressed." David Nicholson (WHOI)	"Impressive. Well done!" Guilherme Castelão (SCRIPPS)
"I'm grateful for all the work that's gone into this Python package. I wouldn't be exploring so much ocean data without it! It's such a great tool for the community -- beyond my research, my students are already showing more interest in ocean data and characteristics because I've been able to share these Argo profile plots in my lectures and on social media" Kimberly Wood (Mississippi State Univ.)	"I just tried out the argopy binder and am very impressed! I think this is really fantastic--will be a game changer for the field, making it much easier to access argo data." Ryan Abernathey (Columbia Univ.)
"I used argopy in a class HW this semester and thought it was great!" Jacob Wenegrat (Univ. of Maryland, Atmos and & Ocean Science)	

Figures

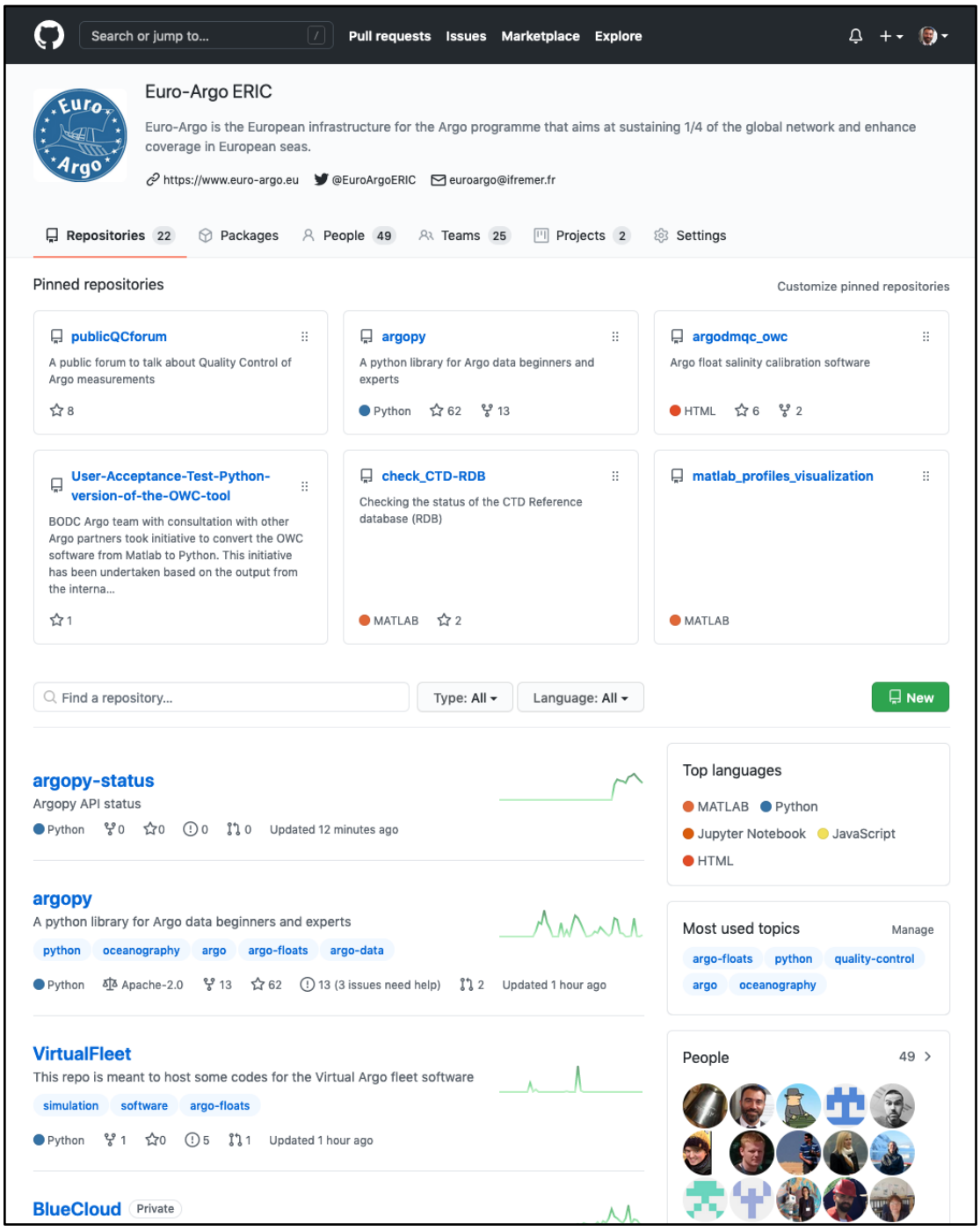


Figure 1. Screen shot of the Euro-Argo Collaborative Framework landing page. Screen captured on Dec. 9th 2020.

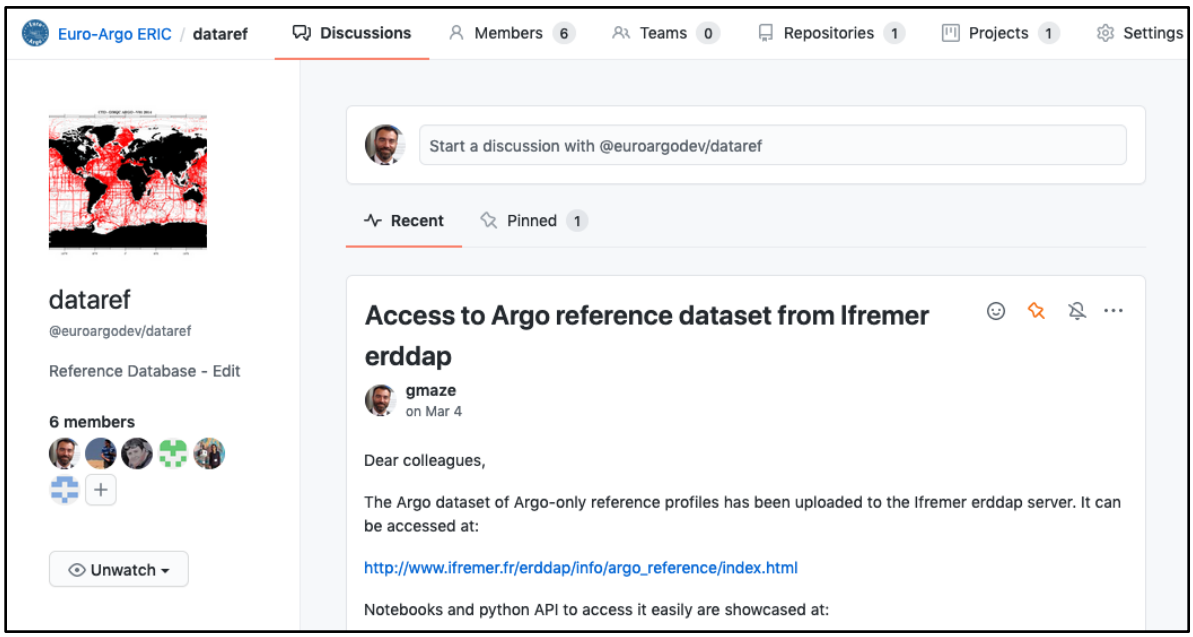


Figure 2. Example of a "team" page showing the team members and a discussion thread only accessible to this team member. Screen captured on Dec. 9th 2020.

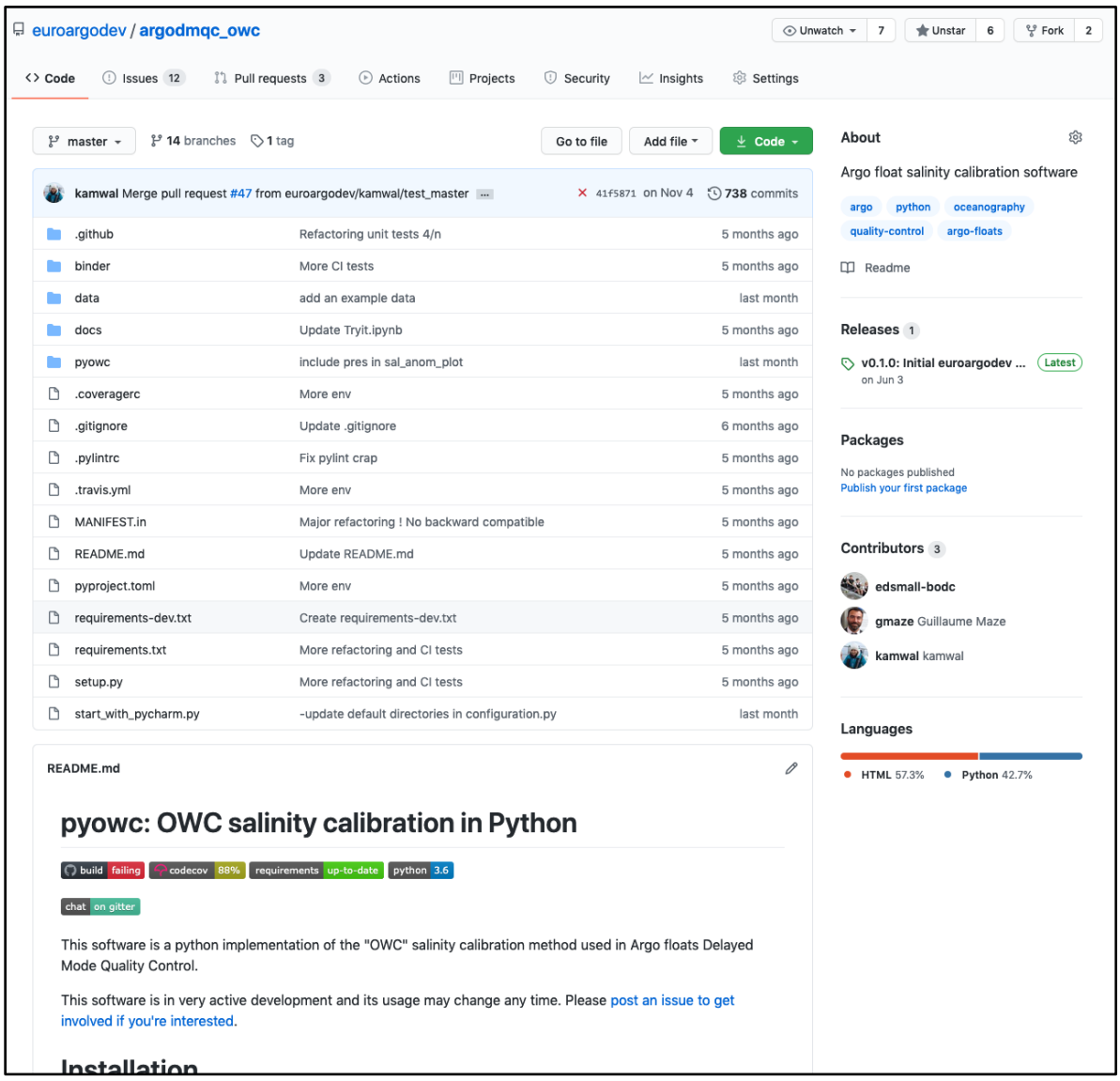


Figure 3. The landing page of a repository where collaborative software development is in action. This is an example for the OWC salinity calibration in python software. Screen captured on Dec. 9th 2020.

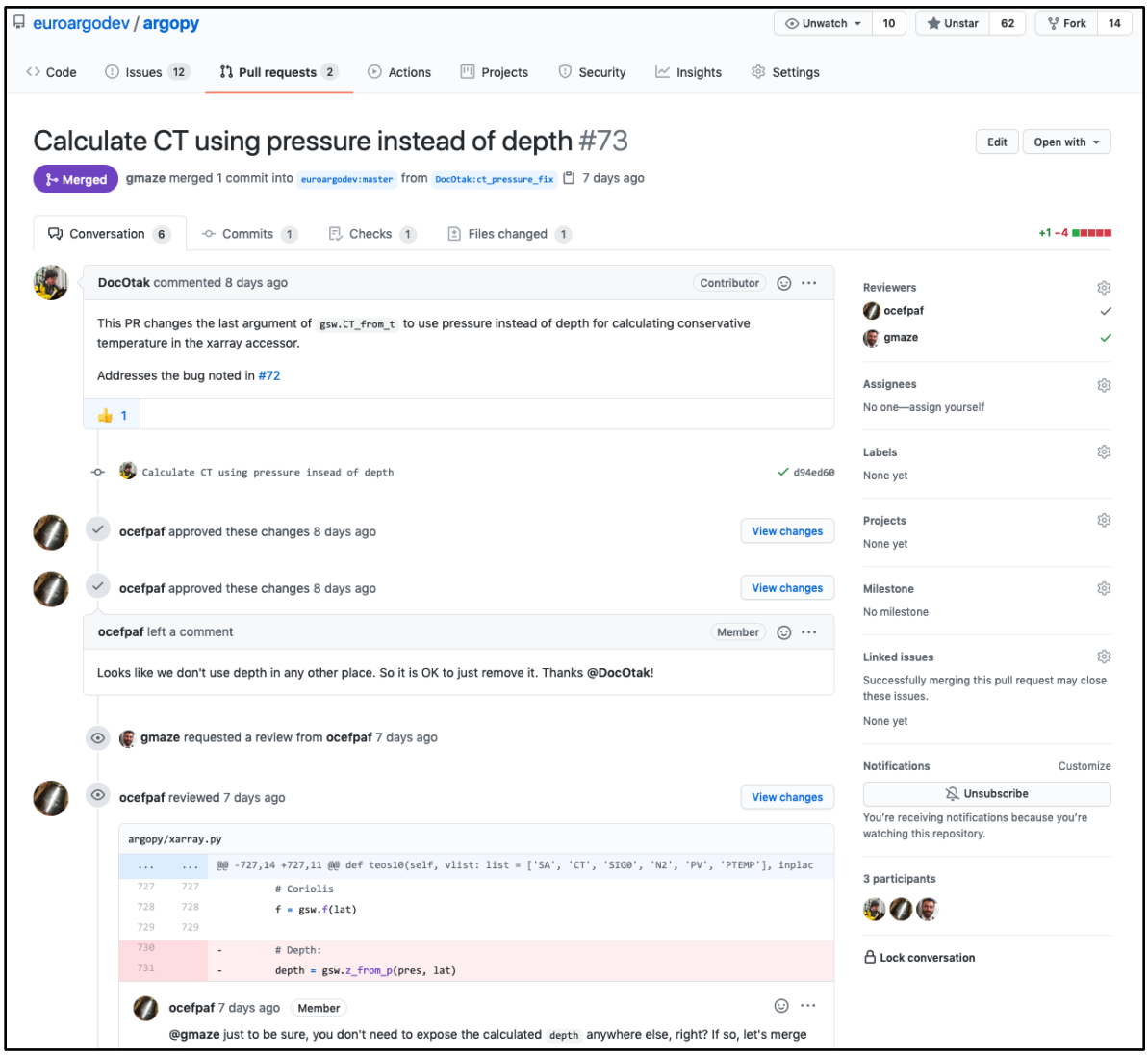


Figure 4. Example of one of the EA-CF web pages documenting technical information and history of a "pull request" (PR): work of a contributor to a software. One can see the title, the contributor's name and description of the work being done within this PR. Each PR can be "assigned" to a one or more collaborators and furthermore request for reviewers that will be in charge for the evaluation of the new code or code changes. Once reviewers have validated changes, one of the owners of the repository can merge the changes to the main code, closing the cycle of contribution. Screen captured on Dec. 10th 2020.

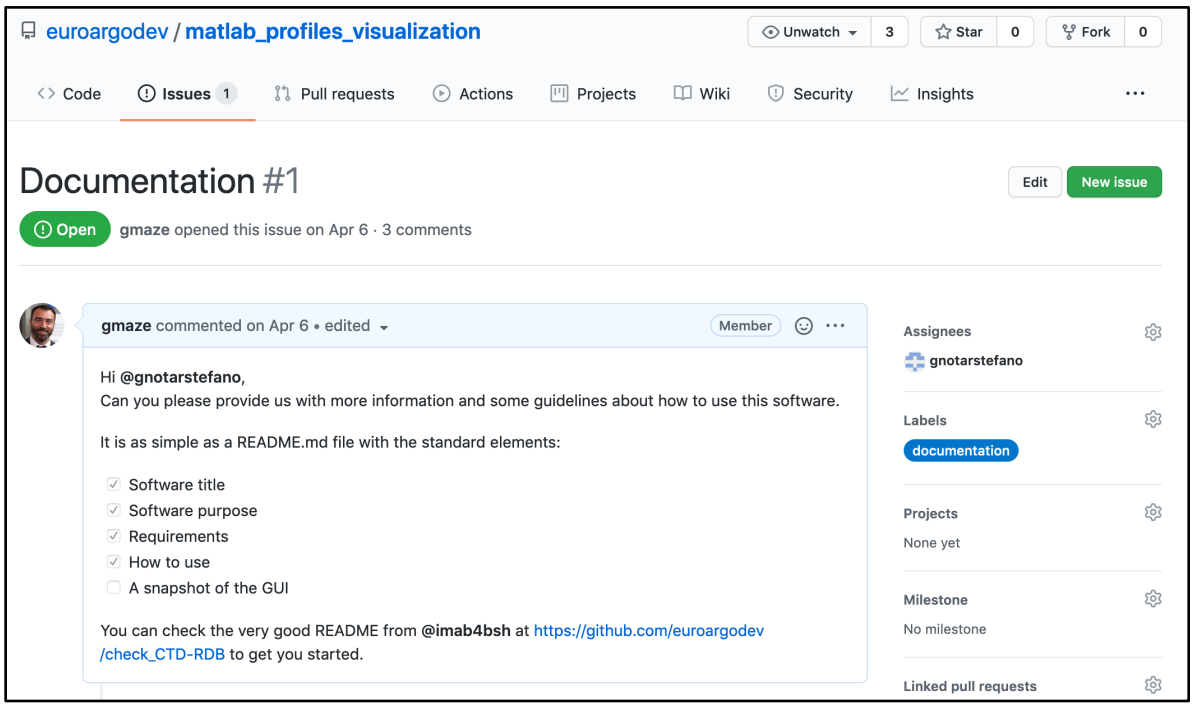


Figure 5. Example of an issue raised on the Matlab profiles visualisation software developed by INOGS. Screen captured on Dec. 10th 2020.

Figure 6. Example of an issue populated with figures (access here). Any type of document can be attached to an issue with a simple drag and drop. If possible, the document content will be displayed, otherwise a link will be created. Screen captured on Dec. 10th 2020.

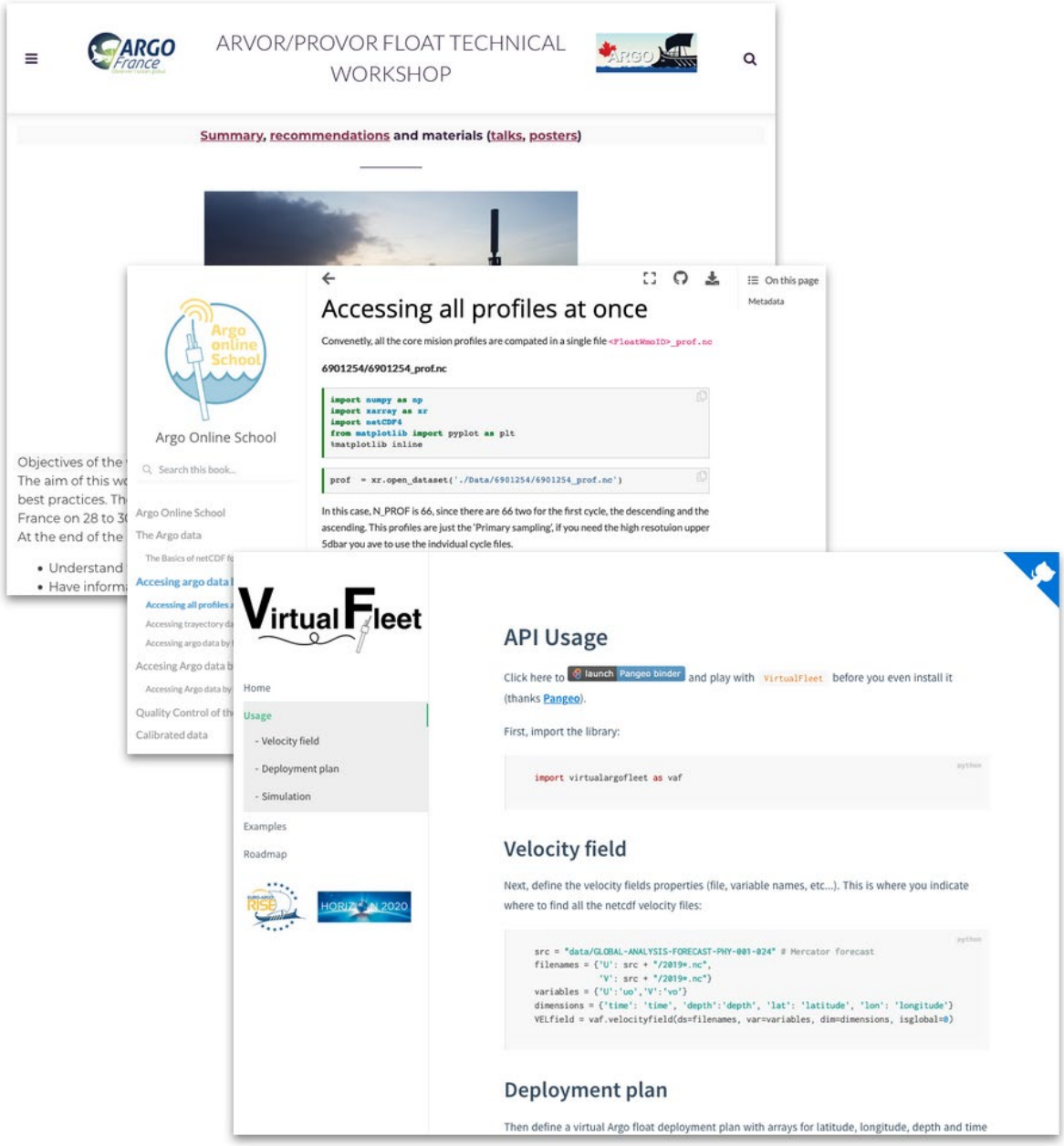


Figure 7. Examples of new websites developed under the EA-CF.

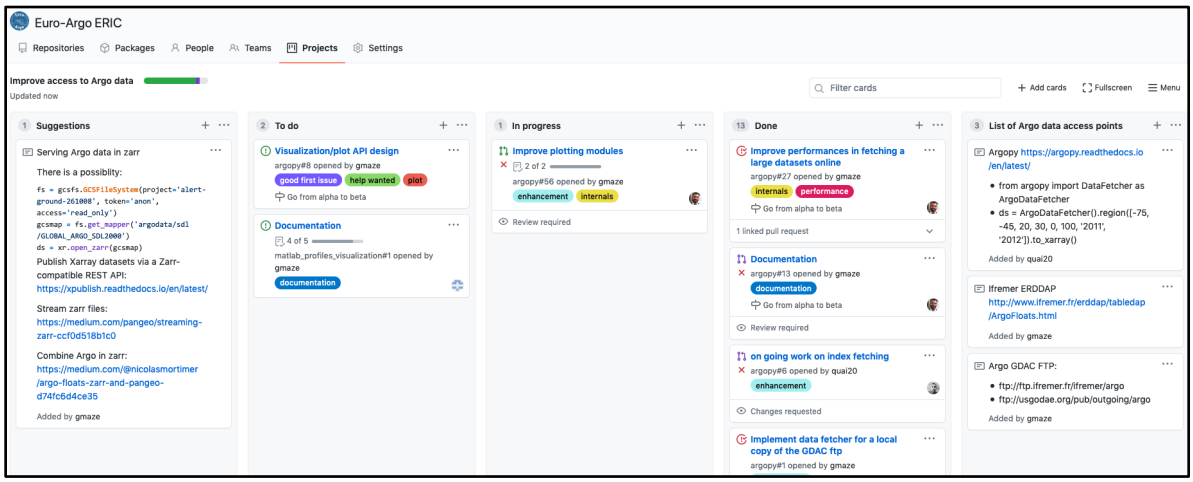


Figure 8. Example of a project board. Project Management boards are available at the organisation or repository level. Screen captured on Dec. 10th 2020.

Figure 9. First training session organised by IFREMER to help Euro-Argo RISE partners get started and to adopt the EA-CF. The session was open to all Euro-Argo RISE Work Package participants of the General Assembly to promote the EA-CF and to prepare for the enlargement of its core users. March 3rd, 2020, Paris.

Figure 10. Presentation of the EA-CF components to the ADMT21 online meeting, which gathered about more than 100 participants during 3 days. Screenshot taken on Dec. 3rd and 4th 2020.

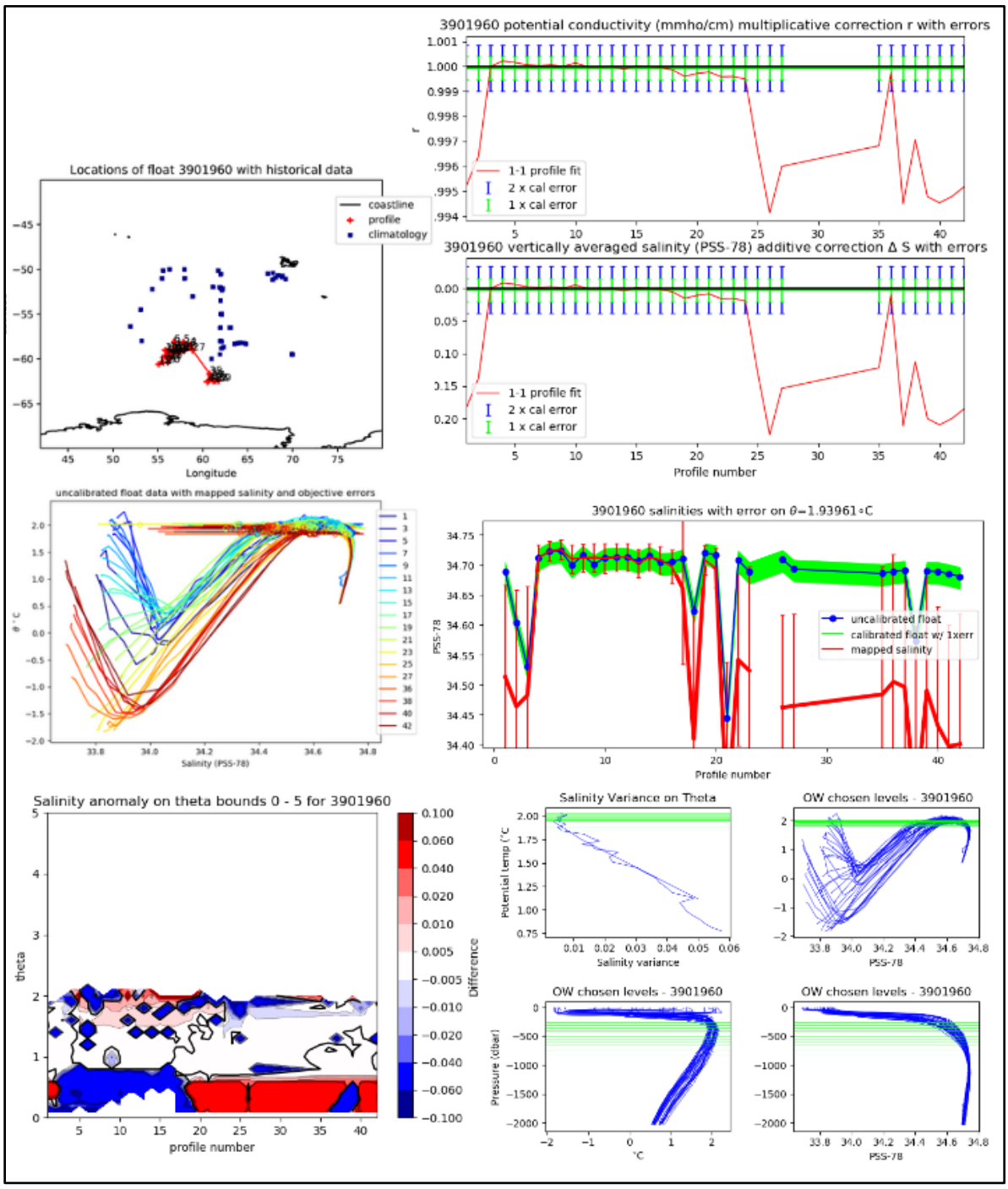


Figure 11. Various graphics showing results of the new DMQC salinity calibration software pyowc.

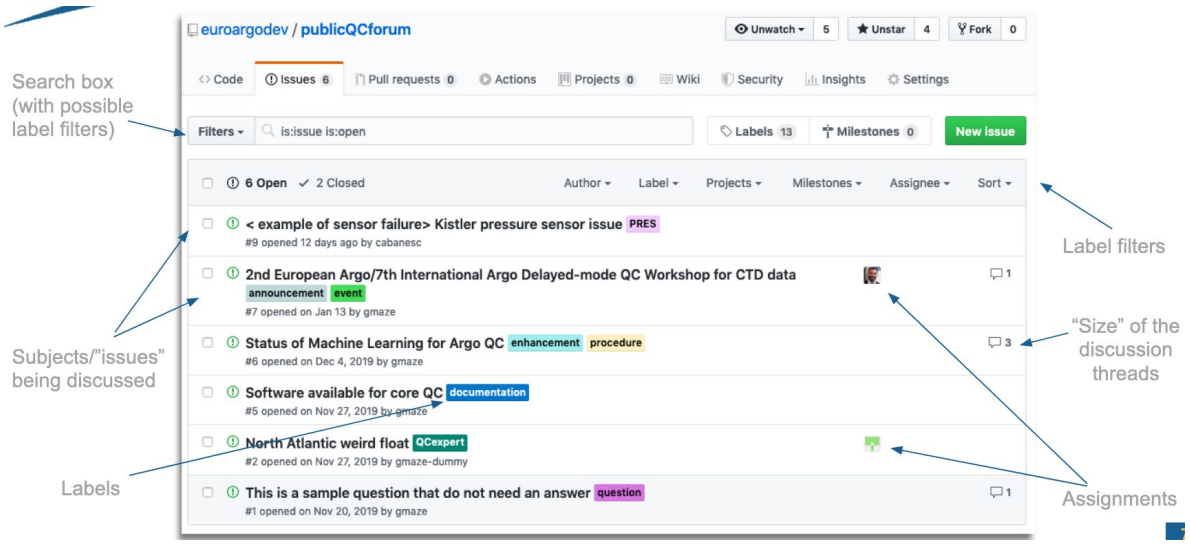


Figure 12. Main page of the Argo QC forum created for the EA-CF. Main components are annotated. Presented during the 2nd Euro-Argo RISE General Assembly, March 2020.

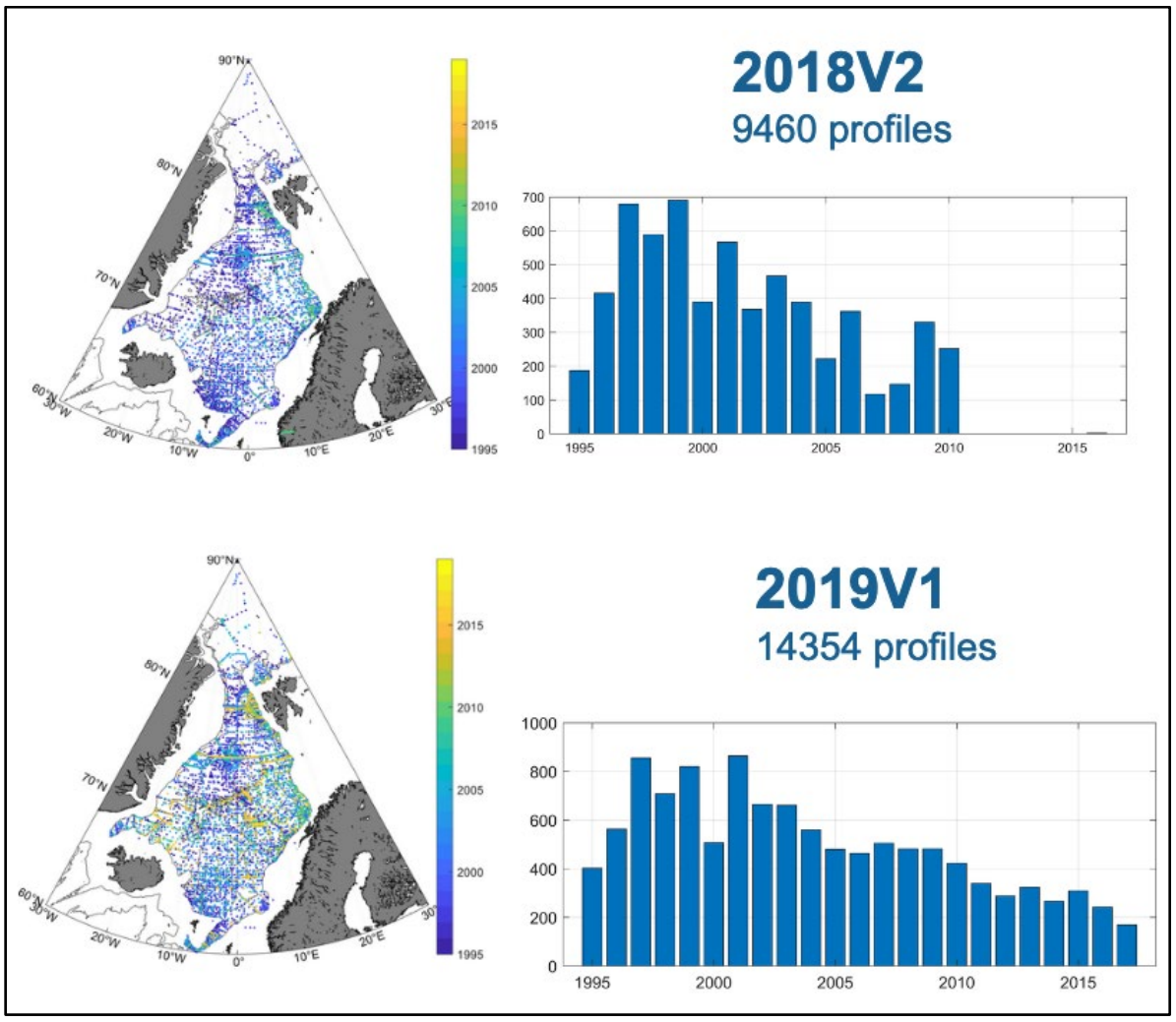


Figure 13. Improvement of the Argo CTD ref-db in the Nordic Seas. The map shows the position of all profiles available and the year of sampling is color-coded. The histogram shows the number of profiles per year.