Capacity Exchange

Technical Plan

Current Situation

The Capacity Exchange's current pilot platform is a derivative of OERWorldmap software, based on Java and Ruby. It has been adapted to use Mediawiki Login. It allows users to create their own profile and search for other users' profiles. This is based on different skill tags and displayed by the date of profiles' creation or edition. Furthermore, it also allows users to pinpoint their exact location on a globe. It could enable searching for editors based on their geographical location.

Among the code's main functionalities, which serve the needs of the second phase of the Capacity Exchange project, are: profile creation, editing, and viewing; access by tag search; and scalability when changing from desktop to mobile. However, geolocalization can be considered quite counterintuitive, given that the Wikimedia Movement is organized mostly based on languages, than it is by countries. Additionally, concerns might arise regarding users' privacy when mapping people geographically.

The OERWorldmap has since been deprecated on Github.

The current prototype is a stable system, but it will be very difficult to adapt to our future needs. This is why this document contains two aspects:

1. Keeping the current software running
2. Working on a new solution and migration scenario

As we will promote the Capacity Exchange in the coming year(s), it is genuinely necessary to make clear to users that there will be a new platform and we need to ensure that the data is migrated (with some criteria).

Keep the current software running

While we keep the current software running it is critical to continue learning from it.

Although we propose switching code language, there is plenty to learn from the current prototype as it is. Based on this, an initial test should target focus groups composed of representatives of outreached Sister Initiatives.

Focus groups are an effective method of usability evaluation in software engineering. Fast and cost-effective, the method provides valuable empirical insights. To test the Capacity Exchange's current prototype we expect to conduct an individual, hands-on exploration of the website, connected to a group discussion of collective perceptions. We can then assess the overall experience and feedback of the Capacity Exchange's key stakeholders. This should guide on-development screens and task workflows. Such an inquiry can validate if what is being created is indeed aligned with the expectations of the potential user base.

Focus groups

The focus groups' participants should be Wikimedians, part of Capacity Exchange's Sister Initiatives. The groups should consist of 3 to 4 participants at a time, and last 90 minutes. The gatherings will be conducted online, from mid-September to mid-October 2023, according to participants' availability.

The Program Manager and Software Developer will design a specific questionnaire. The Manager will also mediate the sessions and run a follow-up assessment, according to the following plan:

Focus Groups		
Duration	**Activity**	**Description**
5’	Introduction	Presentation of the session dynamics (the participants will already be acquainted with the project and its current prototype, from previous meetings with the Program Manager).
15’	Exploration	A script of possible tasks the participants may perform will be presented (create profile, fill up profile, navigate the map and skills tree, find individuals and groups with a specific skill). However, they are free to explore the website. If they face questions or difficulties, they need to explore as if they do not have outside assistance.
20’	Questionnaire	Immediately after participants explore the website, they will individually and anonymously answer a short questionnaire.
50’	Discussion	The discussion will center on the questionnaire results (instantaneously summed up by specific survey tool); and will enable participants to provide overall comments regarding their experience in the guided explorations; as well as their remarks on the website look, feel, and functionalities.
Assessment		
Based on the questionnaire and discussion results, the Program Manager will reach the focus group’s participants asynchronously to analyze and validate the current skills tag framework. This will be done according to each Initiative’s capacity building work. Moving such an assessment of the focus groups’ synchronous dynamic to an asynchronous follow-up activity is a strategy to make the sessions shorter, thus increasing the likelihood of participation and enabling more in-depth commentary regarding skills tags.		

Within the development timeline, the focus groups are part of both testing and feedback activities, respectively marked in green and yellow.

Work on a new solution and migration scenario

After diagnostics, we concluded that the future platform would best be served by a newly developed infrastructure and code language.

The language switch from Java to Python enables the Capacity Exchange's online longevity and sustainability. The aim is to leverage the larger Python community within the Wiki movement. As a result, a larger pool of developers will be able to contribute to and maintain future additions – whether features or bug fixes.

Analyzing data from Toolforge further strengthens the case for Python, as it demonstrates widespread language usage within the hosting platform. Out of the hosted tools, 43 mention "python" as one of their tags, compared to only one tool mentioning "java". While it should be noted that many tools haven't listed their tags or descriptions on Toolforge, this still points towards Python's widespread adoption and community support. Additionally, focusing on the two Python frameworks under consideration – Django and Flask – their presence on Toolforge is substantial, with 6 and 15 apps tagged, respectively. This reinforces Python language familiarity. It also highlights the advantage of selecting a framework with a strong presence among community developers.

Transitioning to Python is a strategic action that aligns with the Wikimedia Movement. It takes advantage of the language's widespread adoption and robust frameworks. By doing so, CapX would benefit from increased contributions, a larger pool of potential developers, and an increased likelihood of long-term maintenance and success. Moreover, embracing Python will empower us to craft screens, user interfaces and experiences that would otherwise be challenging to achieve. This is while building upon the current Java prototype. It will allow us to create a better peer-to-peer connection experience for all users.

Costs

The switch to Python entails a sizable cost in terms of time and effort, as it requires rewriting sections of the code, and writing many others from the ground up. Such an undertaking would require substantial development effort, distinguishable between:

- **Development** tasks are those pertaining to functionalities that don't exist in the current prototype. These are sections of code that would have to be developed even if we continued to work on the current prototype.
- **Migration** tasks are those related to functionalities that already exist, even partially. These are sections of code that already exist and would have to be rewritten in Python. These are the ones we will focus on in this section.

In the subsequent section outlining the development timeline, migration and development tasks are respectively marked in red and blue.

Rewriting the application and ensuring the functionalities in a new language should consider the following:

1. A thorough analysis of Java code is essential to identify functionalities and dependencies that must be replicated. Planning the conversion also includes mapping the Java structures to their Python equivalents.
2. Some of the functionalities in the prototype – such as the map displays – might not be immediately replicable in Python, due to differences in the libraries or modules used to achieve them. It will be necessary to identify Python libraries that are best suited to achieving the same functionality.
3. This also entails a change in documentation, as well as writing tests to ensure the code works correctly. It will be necessary to write tests to guarantee that these migrated functions work in both regular and corner scenarios.

4. Several functionalities in the current OERWorldmap software are not used at all, and have no role in the future. It is a must to identify the functionalities that have to be migrated, and only migrate those.

Rewriting the current prototype would be time-consuming. However, it would also allow us to more readily implement improvements based on what we have learned from tests with the current prototype, as well as future learnings from the further testing we intend to do. It would allow more flexibility in programming, which would result in better end product usability.

Project Development Timeline

The Capacity Exchange Activity Timeline covers structural development (2023-2024) and iterative development (2024-2025).

Initially, the timeline covers a migration phase: testing the current prototype, and studying the final product. The migration cost will be the weeks spent studying the next steps of a Python implementation. Furthermore, Java code parts will have to be reworked in Python. These are marked in red on the timeline, and it is estimated to take 6 weeks in total.

In the Activity Timeline, the development phases are marked in blue, and cover the majority of the first year timeline. They encompass some functionalities that already exist in the current prototype – such as the profile and search systems - since those would have to be reworked to achieve our vision for the product. This phase is estimated to take 35 weeks.

Functionalities in Django are developed as apps, which work as complete sections of code independent of one another. Each app implementation phase will include:

1. Reviewing the app's exact features;
2. Reviewing the intended taskflow;
3. Implementing the functions to make them work;

4. Implementing thorough tests covering at least 80% of the code;
5. Documenting the functions.

The project will be built on the Django framework. It allows for rapid development, abstracts database operations into Python and its admin interface makes data management easier. It has an active community, even within the Wiki – some Toolforge Python projects also include the Django tag.

Of the functionalities mentioned above, we stress that reporting issues and suggesting possible new features should be an easy process for CapX users. It is crucial to ensure that even those who are not well-versed in software development and collaboration tools like Github or Phabricator can easily report bugs and suggest features. To achieve this, the chosen method for submitting such requests will be a user-friendly bug reporting app directly on CapX's website.

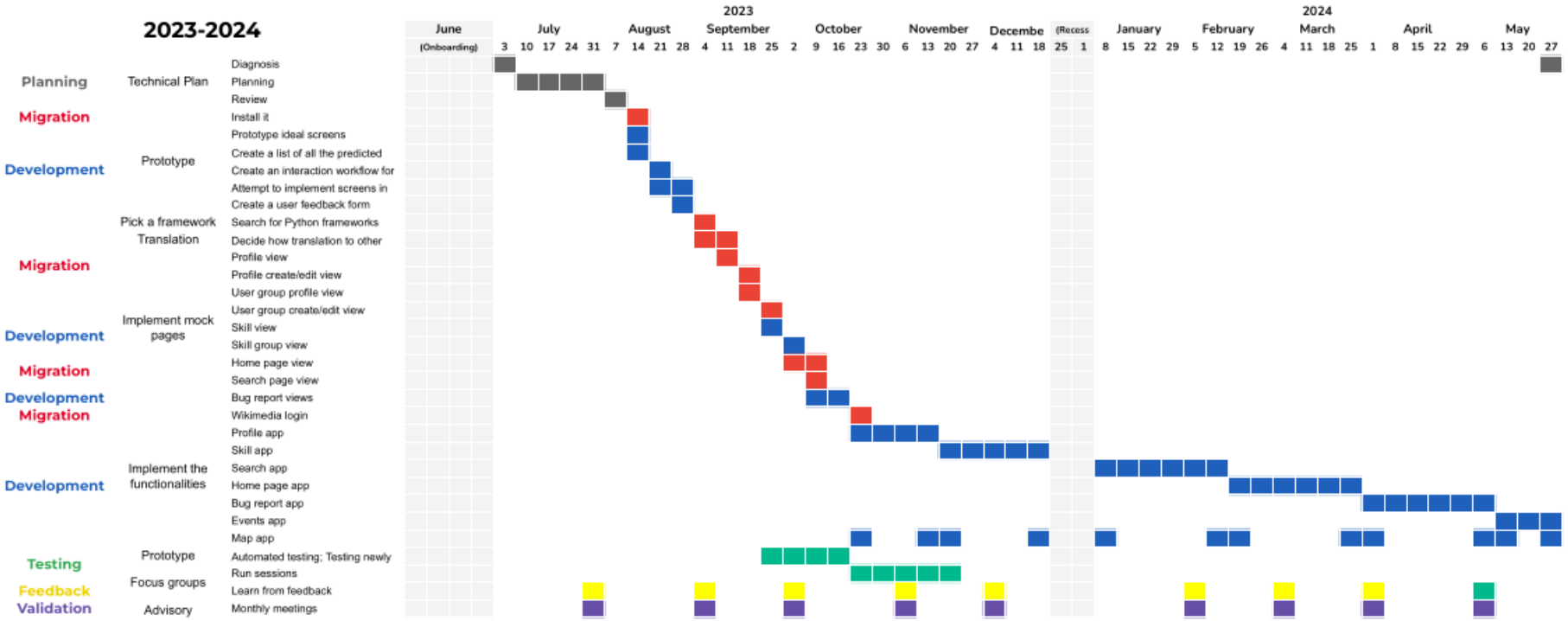
The bug reporting app will offer a simple and intuitive form that allows users to submit feedback effortlessly. Users will have the option to select the type of issue they encounter. They will also be able to provide a clear and concise description of the problem or feature they'd like to see added.

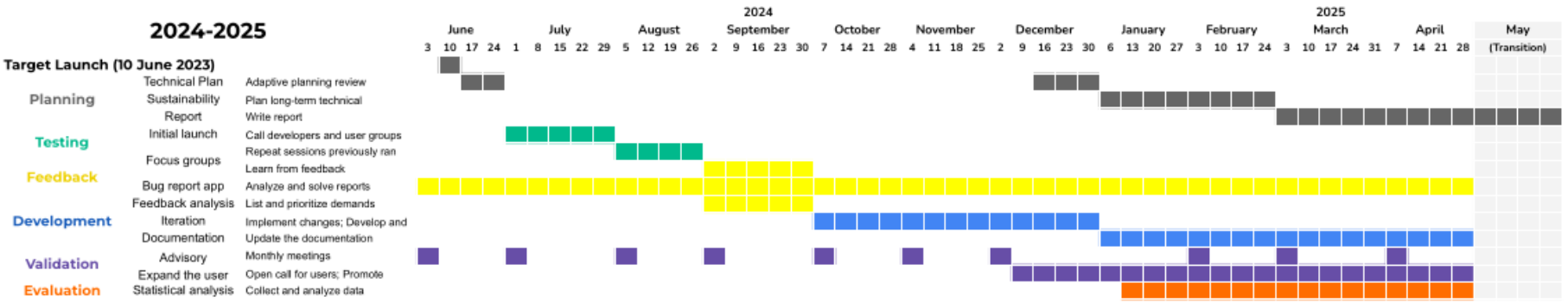
To promote transparency and avoid duplication of efforts, the app will allow users to view previously reported issues. They will be able to keep track of the status of each report, whether it has been fulfilled or not. This will ensure they stay informed on whether their concern has been addressed or is still pending resolution.

As for documentation, all functions in the code will be documented, including their purpose, what variables they receive and what values they return. Detailed documentation will be done upon the completion of different sections of the app, to avoid redundant work on this text. This is as the functions get further refined. Comprehensive tests will serve as additional documentation. If community interest arises, it is also possible to write a more thorough and non-programmer friendly documentation version. The documentation will be done through readthedocs.io.

Aside from automated testing, throughout development, we expect to test and gather feedback from users with different levels of involvement with the project.

Activity Timeline





The second year of project development will cover iteration processes such as various activities of product testing, feedback and statistical analysis, incremental development, and documentation updates.

Our target launch date is 10 June 2024. This is when the project will have completed one year of development and all the main migration and development activities.

The planning activity will consist of an adaptive review of the present plan. This will accommodate unexpected changes and needs that might arise during the first year. This also includes the planning of the Capacity Exchange's long-term technical sustainability and governance. By the end of the second year, this activity will also encompass report writing.

Capacity Exchange aims for an agile and participatory methodology, which includes planning activities, as observed in the present document.

After the product launch, we'll call developers and target user groups to test it. In 2024, we'll re-run the focus groups of 2023, comparing feedback from previous and current prototypes.

We'll then learn from feedback, analyze and solve reports. In light of this, demands should be listed and prioritized before an iteration begins. The iteration will implement needed changes, as well as new features.

Documentation, as mentioned before, is a crucial aspect of this project. It will also be carried out during the second year of project development.

All the listed activities will be validated through periodic consultation with the Advisory Committee. By the end of the second year, this will also include an open call for users and workshops. It is imperative to expand the user base, from Sister Initiatives to the wider Wikimedia Movement. This is to better understand different perceptions and needs regarding product functionalities.

Finally, we'll evaluate product usage and feedback, aggregating this data into the project final report.At last, we'll evaluate product usage and feedback, aggregating this data to the project final report.