

Elsevier Pure: Development of the system

Kristianstad University (HKR)

August 2020



Table of Contents

1. Introduction.....	2
2. About Elsevier.....	3
Research Intelligence	3
Pure.....	3
3. Development processes and tools	4
Version control	4
4. Release Management	6
Release testing	6
Continuous Integration.....	7
5. Approach to long-term development.....	8

1. Introduction

This document provides a response to requirement 1.4.5 Development of the system in Tender 20/2 for a Current Research Information System, which states:

The tenderer must have an approach to guarantee quality over the release- and development cycle for both existing and new functionality (development of new features, quality assurance before release) and an approach to the system's long term development including the influence the customer will have on the development

2. About Elsevier

Elsevier is a global information analytics business that helps institutions and professionals advance healthcare, open science and improve performance for the benefit of humanity. Our products are used at more than 25,000 Academic and Government institutions around the world. We have a long history of collecting, structuring and delivering research information to a global audience, and we continue to invest in technology to further enhance our ability to do so. For example, we employ 1,000 technologists and invest \$400m annually in technology. We also have great focus on hiring and training next-generation data scientists to help us evolve with our customers.

Every tool we develop is built on trusted information — because from the library to the lab, from the institution to the individual, scientific and medical advancement requires reliable data. At Elsevier, we have partnered with the scholarly community for 140 years to curate and verify knowledge. Now, as an information analytics business, we are committed to bringing that rigour to a new generation of Elsevier platforms.

Elsevier is a global business headquartered in Amsterdam with offices worldwide. We are part of RELX Group, which has about 30,000 employees based in 34 countries across the world, working across six continents and serving customers in more than 180 countries.

Research Intelligence

Elsevier's **Research Intelligence** portfolio provides solutions for each stage of the research lifecycle. Our portfolio spans research workflow tools, information management, repositories, research data management, and extends to powerful tools for open educational resources, open access publishing, and support for the full spectrum of scholarly and creative works. Our vision and work strongly support easy access to research data as a cornerstone of success in modern science, and we help individuals and institutions use data to its fullest potential across the researcher workflow.

Pure

Pure is Elsevier's market-leading Current Research Information System (CRIS). It is one of the cornerstones of our Research Intelligence portfolio. As a versatile, interoperable software solution, Pure can be configured to meet the growing requirements of your organization. Pure's industry-proven data model unearths multifaceted insights about the overall research lifecycle, supporting both fact-based decision-making and industrial-strength expertise discovery. It aggregates your research information from numerous internal and external sources, ensuring the data driving strategic decisions is trusted, comprehensive and accessible in real time.

Used by over 260 institutes of education and research globally, Pure is the market leading CRIS. A list of Pure customers can be viewed here:

<https://www.elsevier.com/solutions/pure/clients>

The following sections describe how we assure quality within the Pure development process, including both during and after release.

3. Development processes and tools

We use Agile principles and methodologies in our development of Pure. As with other Agile software development teams, we collect and maintain requirements in different ways. In our case, we use four tools:

- **Jira:** Our issue and project tracking system. We use Jira to capture and maintain requirements. In certain cases, we invite clients to Jira, so they can help us define requirements.
- **Agile principles:** Our developers, product managers and UX team use the Scrum methodology to manage requirements and ensure our backlog is prioritized by importance and urgency. In certain cases, clients are also included in these discussions.
- **Confluence:** Our collaboration wiki. We document more comprehensive requirements in Confluence. In certain cases, we invite clients to Confluence, so they can help us define requirements.
- **Informal mock-ups:** Usually produced by our product managers, UX team, developers, and in certain cases, clients.

We use Jira to manage and track both client-reported issues and any issues found by our QA teams at Elsevier. Issues are firstly sent to the Pure Support team who identify relevant stakeholders and triage whether issues can be solved by the Support team or sent to relevant Engineering teams. Jira workflows ensure that the status of each issue can be tracked and is visible to the reporter. These workflows also allow the reporter to re-open an issue if they are unsatisfied with the solution.

Version control

We use Git as our software version and revision control system. All changes and modifications done as a part of development, support, and maintenance of Pure are registered through Git. This applies to all types of Pure release. For more details on the different types of Pure release, see [4. Release Management](#) below. The tight integration of Git enables us to track all changes and make reliable references to all customer releases and issues reported. Git is also used to maintain Pure configuration.

Git is used to support our quality assurance procedure by branching and tagging software. A typical scenario (as illustrated in Figure 1 below) is:

- Developers commit all new work, such as new features and bug fixes, to the main development branch (master) on a day-to-day basis.
- When a new major release is planned, for example 5.14.0, our development team creates a release branch by copying or branching the master development branch. In the example in Figure 1, this release branch is 5.14.0-SNAPSHOT.
- A dedicated team of developers and testers (known as the verification team) then start the test and verification period. They branch '5.14.0-SNAPSHOT' into '5.14.0-RELEASE' and then close that branch, which leaves master and '5.14.0-SNAPSHOT' open for the remaining teams to continue development, where they will not disturb the verification team.

- If bugs are discovered, fixes are ported back and forth as necessary, and also ported to '5.14.0-RELEASE' if the verification team determines it as critical for the release.
- Upon release, we tag '5.14.0-RELEASE' in Git as '5.14.0' for reference. This tag is used when distributions are built and packed for customers. The '5.14.0-SNAPSHOT' is maintained in parallel with master and bug fixes continue to be ported from 'master' to the snapshot branch.
- Around the first of every month, a maintenance release is planned and '5.14.1-RELEASE' is created and tested before it is tagged as '5.14.1' for distribution to customers. The maintenance of '5.14.0-SNAPSHOT' continues until the next feature release (5.15.0) is released, after which it is closed for further development.
- Throughout the maintenance of '5.14.0-SNAPSHOT' a hotfix/emergency release can be made when needed. A hotfix/emergency release is based on the latest maintenance release (5.14.1) by porting fixes from '5.14.0-SNAPSHOT' to latest release branch (5.14.1-RELEASE). The hotfix is then tested on the release branch and tagged as e.g. '5.14.1-1'.

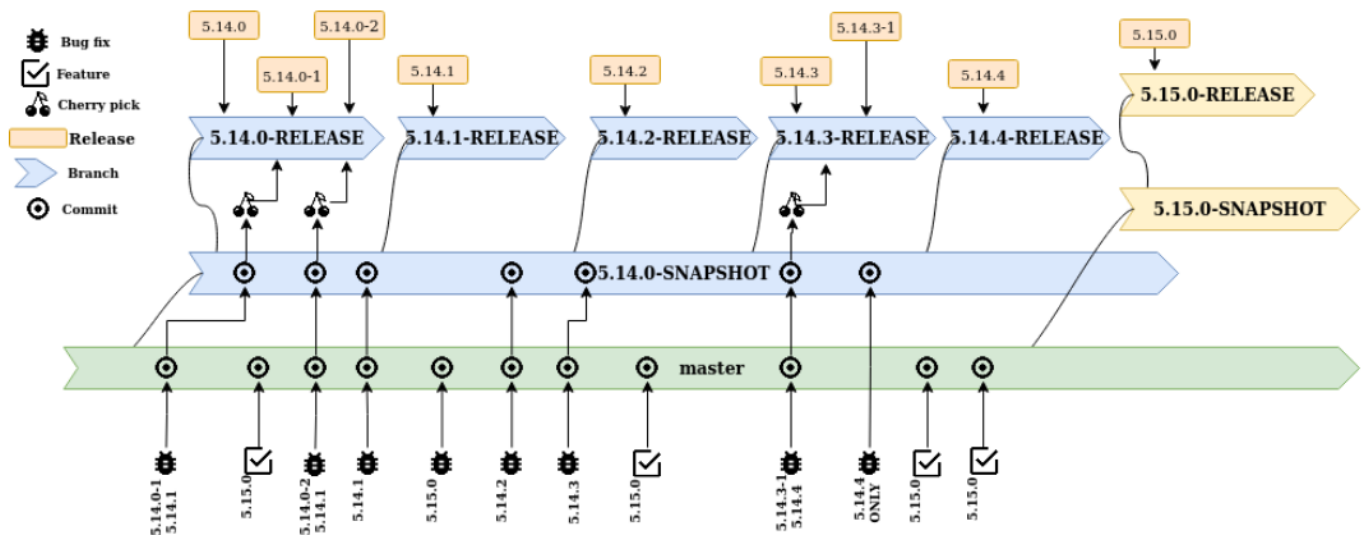


Figure 1: Pure release cycle.

4. Release Management

We have three types of Pure release in order to keep up to date with our clients' needs:

Release type	Purpose	Schedule*
Major release	Contains new features and the most recent bugfixes. It also includes all minor releases.	Three times a year — usually in February, June and October.
Minor release	Contains bug fixes and feature enhancements. We do not maintain minor releases after the following major release — for future improvements, you should upgrade to the next major release.	Around the first of every month.
Dash release	Released to support an urgent fix to a version of Pure.	Only when needed.

We publish release notes for Major and Minor releases of Pure on Confluence, our collaboration wiki. These release notes contain information on new features, resolved issues, any special requirements and other relevant information that clients need to know before upgrading Pure. We also send an announcement to clients via email about upcoming releases.

We involve our user community in the shaping of our development roadmap, as this helps us to better understand our end users' needs. These developments include not only new features, but also improvements in the services surrounding the implementation and maintenance of Pure. The Pure Product team determines the priority of these development requests, based on factors such as client demand and capacity of the development team.

Release testing

Before a release a team, usually consisting of Developers, Testers, and a Test Manager, is assembled. The team is dedicated to the verification phase until the software is released. The software is branched out to a release branch. The team is given exclusive right to the branch and is in full control changes going in or out. This is to stabilize the software, increase the knowledge of the release, and minimize the risk of introducing new issues.

One of the first activities of the team is to identify functionality and features that have undergone larger changes or were added since the last release – also called a risk analysis. Along with input from ongoing projects and team experience, we make a list of potential focus areas and test subjects for the team.

A verification phase is usually divided in three sub phases. The first phase is a combination testing and issue fixing. Identified test subjects are tested and any found issues are reported to Jira, fixed and committed to the release branch. During the second phase, important and business critical functionality of Pure are tested and solved issues are verified and closed in

Jira. This is repeated throughout the phase. The third phase is to gain as much knowledge and confidence to release as possible and log any issues in Jira.

Continuous Integration

At Elsevier, we have Continuous Integration (CI) testing integrated in our infrastructure, which means that committed changes are immediately tested and reported on when they are added to our code base. CI is used to provide rapid feedback so that if a defect is introduced into the code base, it can be identified and corrected as soon as possible. Because CI detects deficiencies early on in development, defects are typically smaller, less complex and easier to resolve.

Every night we perform even more in-depth testing on the latest committed code. The results are summarized in a test report, and if necessary, issues will be sent to the developers.

The results from CI and our nightly tests are also taken into consideration when releases are tested.

5. Approach to long-term development

Our clients contribute to the development roadmap through their local country-level Pure user group. They also have the option of raising new feature requests or suggestions to existing functionality through Jira, our issue tracking system. The development roadmap is shared with the user groups, so they always know what is coming; both in near-term releases and over time as a result of the product strategy.

Pure's roadmap is continuously evolving to anticipate our client's future needs. Over the coming years, Pure clients will benefit from:

- Support for additional funding office workflows with more flexible options. For example, further support and tracking for pre-award and post-award grant management workflows.
- Data administration activities with reduced effort due to:
 - Increased coverage of research data sources.
 - Expanded API framework allowing bulk content editing.
 - Increased use of Elsevier assistive AI technology for recommending potential content to fill gaps in the systems' content.
- Enhanced new features for showcasing an institution's expertise externally to targeted audiences.
- Improved reporting/dashboard capability from within Pure (more out-of-the-box reports).
- Facilities for collaborating institutions to share researcher profiles for transferred faculty.

Interoperability will remain a priority for institutions, and in response, Pure will further continue its support for international standards of research data exchange between systems, institutions and with funders, e.g. OAI, various versions of Dublin Core, CERIF. The flexible Pure Web Service API on top of our robust metadata model ensures that clients can exchange interrelated high-quality content with assessment stakeholders.

Globally, we see an increased demand from funders for detailed tracking of open science requirements. We will continue to expand Pure to help institutions gather in-depth data and perform the tracking necessary to comply with research funder mandates and share evidence of that data to necessary parties. Funders are increasingly requesting that research teams report on the “beyond-academic” real world impact to society, the economy or patients. Pure will continue to expand its present Impact Module and range of social impact metrics to help universities tell their impact story to funders.

Finally, as management of systems becomes more intelligent, Elsevier plans to invest further resources in Natural Language Processing and machine learning (AI techniques). This will lead to more efficient processes for clients; from getting content into Pure with less effort or more intelligently matching funding opportunities to talent you have in your organisation.