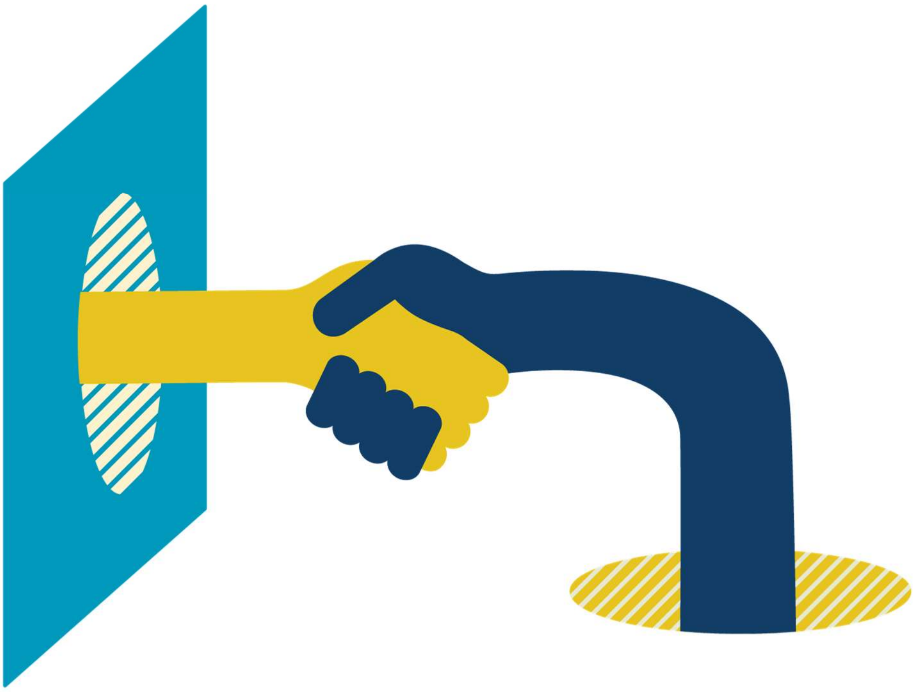
Jems
Capitalisation Report

June 2025

This document capitalises on the 5 years development process of Jems and captures important lessons learned for projects with a similar context.



Disclaimer: Cooperation can be complex, and while Interact's job is to make it easier, Interact cannot offer assurances on the accuracy of our pan-European information in any specific context.

Furthermore, understanding and knowledge evolves throughout the programming period. If you spot a something out of date or inconsistent, please contact us at communication@interact.eu

Publisher Interact Programme
Date 12.06.2025
Author(s) Alexandra Kulmer, Jose Almeida, Pieter Louwers

Interact

Co-funded by
the European Union
Interreg

Table of contents

Executive Summary

The Jems Capitalisation Report captures key insights and lessons learned from the five-year development process of the Joint Electronic Monitoring System (Jems). Designed to support Interreg programmes in launching calls, assessing, selecting, monitoring, and paying projects, Jems serves as an adaptable and future-proof monitoring system for the 2021-2027 programme period and potentially beyond.

Key Achievements of Jems as a software:

- **Over 50% of all Interreg programmes use Jems** (indicated by the Jems license holders via the Jems License Holder surveys (2023 and 2024))
- **15.3m Euro saved by Interreg in software development costs** (based on analysis done via the Monitoring systems network comparing with all Monitoring systems used by Interreg first published in 2023 and annually updated)
- **80% of beneficiaries find Jems easy to use** (indicated by the Jems end users via the Jems User Feedback survey (2024))

Background and Rationale

Jems was developed as a successor to the 2014-2020 eMS (electronic Monitoring System), addressing critical limitations in adaptability, user experience, system performance, interoperability, and quality assurance. The need for an agile and scalable monitoring system prompted a ground-up redevelopment process incorporating stakeholder feedback and iterative improvements.

Key Learnings and Challenges

- **Stakeholder Engagement:** Strong collaboration with Managing Authorities, Joint Secretariats, end-users, and the development team ensured a user-centric design.
- **Agile Development Approach:** Adopting agile methodologies enabled iterative releases, continuous feedback loops, and adaptability to evolving regulatory and functional requirements.
- **Governance and Project Management:** A structured governance model provided oversight, strategic direction, and early validation of system requirements.
- **Development Phases:** The project followed a phased approach - definition, core development, additional development, and ongoing maintenance - ensuring a stable and continuously improving system.
- **Quality Assurance and Testing:** Automated and manual testing protocols were implemented, significantly enhancing system stability and reducing bugs compared to its predecessor, eMS.

- **User Support and Communication:** A multi-channel support system, including a dedicated helpdesk, user manuals, and community engagement initiatives, facilitated effective stakeholder interaction and troubleshooting.

Impact and Outcomes

- **Enhanced System Performance:** Jems demonstrated improved scalability, stability, and user experience compared to eMS.

- **User Satisfaction:** Surveys conducted in 2023 and 2024 indicated high satisfaction levels, with increased engagement and ownership among users. A community-driven approach, including a democratic voting system for feature prioritization, ensured relevance and alignment with programme needs.

- **Future Viability:** While current funding and support are secured until 2029, discussions on long-term sustainability and governance models are just starting.

Conclusion

The development of Jems represents a significant advancement in the digitalisation of Interreg programme management. By leveraging agile methodologies, stakeholder collaboration, and continuous learning from past experiences, Jems sets a benchmark for future monitoring systems. Ensuring its long-term sustainability and adaptability remains a priority for the Interact Programme and its stakeholders.

1. Introduction

In this document we aim to capitalise on the important lessons learned throughout the 5-year development process of the Joint Electronic Monitoring System (Jems). Jems allows Interreg programmes to launch calls, assess, select, monitor and pay projects and respective beneficiaries as well as report to the European Commission. Jems is built for the programme period 2021-2027 and, hopefully, due to its high adaptability, can be used beyond this period depending on the changes that might come with new regulations.

Jems is a newly developed monitoring system, built after the success of the 2014-2020 programme period monitoring system eMS (electronic Monitoring System). The reason why Jems was developed from scratch, was that eMS was not built in a way that it could be easily adapted to the changes required because of the adoption of the new regulations. Important lessons that were learned from eMS were:

- **Communication and transparency:** During the development process, communication and transparency with stakeholders should be improved.
- **User experience:** the software needs to be more intuitive, easier to navigate and workflows should be simpler.
- **System performance:** the system should be better at handling more users working in the system at the same time at important moments, e.g. just before call deadlines.
- **Interoperability:** the system should allow for interoperability and provide interfaces (bi-directional) with other external systems.
- **Quality assurance:** the system should have a higher level of quality (lower number of bugs) through persistent layers of testing (including automated testing).
- **Future proof:** code quality should be reinforced, the architecture and the frameworks should be re-usable for future programme periods.

Through the agile way of working many retrospectives have taken place continuously trying to improve the development process. Besides that, numerous events have taken place engaging different stakeholders. By the end of the development phase, the different stakeholder groups had been offered diverse possibilities to provide feedback on the Jems development process and their satisfaction, either through surveys or through retrospectives. This feedback was incorporated in this capitalisation report. The main stakeholder groups providing feedback were:

- Jems license holders via the Jems License Holder surveys (2023 and 2024)
- Jems end users via the Jems User Feedback survey (2024)
- Jems project team (Interact and Cloudflight) in several retrospective workshops (2024)
- Jems Core Group in a retrospective workshop (2024)

2. Project setup

When setting up the Jems project, the lessons learned from eMS were taken into consideration. The eMS impact evaluation provided useful insights that have been supporting the decision making when setting up Jems.

2.1. Governance

The development and implementation of Jems involved a structured governance system ensuring successful implementation and broad stakeholder participation:

- **Monitoring Committee** – Provided strategic oversight and high-level decision-making.
- **Steering Group** – A sub-group of the Monitoring Committee, tracking project progress and identifying early risks.
- **Hosting Institution** – The City of Vienna (MA27 European Affairs), legally responsible for contractual obligations with the IT provider.
- **Interact team** – conducted the day-to-day management of the project, commissioning and overseeing the development work done by the IT Provider, responsible for the definition of software requirements, prioritization and testing of features, as well as managing the communication with different stakeholder groups.
- **Cloudflight (IT Provider)** – The contracted software development company conducted the development of Jems, responsible for coding, system architecture, and ongoing improvements.
- **User level requirements groups**
    - **Core Group** – Composed of experienced Interreg programmes, involved in defining requirements, prioritizing features, and validating system functionalities.
    - **User Group** – A wider community of Jems users, contributing feedback via bi-weekly sprint reviews, milestone events, and thematic discussions

This participatory model ensured that Jems evolved according to user needs while maintaining alignment with regulatory requirements and programme workflows.

2.2. Project steps and milestones

When in 2019 Interreg Programmes raised the wish to work on a joint electronic monitoring system for the 2021-2027 funding period, Interact commissioned Ernest & Young to conduct a feasibility study including a risk assessment, as well as an estimation of the required budget and development time. Based on the results, the Interact team developed a project plan including goals and objectives, time and budget. On this basis, the development of the new joint monitoring system Jems was endorsed by the Monitoring Committee in December 2019.

In 2020, Jems development started following distinct phases to ensure timely delivery of critical features.

2.2.1. Definition Phase (January – April 2020)

- **Definition of project scope with Cloudflight**: From February to April 2020, Interact conducted several workshops with the external software development company Cloudflight to set up the structures for the implementation of the software. These workshops were targeted to define the project scope, roles and responsibilities, communication channels as well as to review use cases and prepare the first user stories. Based on the workshop results, Cloudflight has validated the feasibility of the project implementation within the foreseen budget and time plan.

- **Stock-taking with Interreg Programmes**: In April 2020, Interact conducted a survey among all Interreg Programmes, in order to collect information about the technical requirements for a new monitoring system and to review the functionalities needed by Interreg programmes for the 2021-2027 funding period.

As a result of the definition phase, the project structure including the governance mechanisms and stakeholder roles was defined, the project teams at the side of Interact and at the side of Cloudflight were set up and the first epics and a set of user stories were prepared. Based on these preparations, the core development phase could be started.

2.2.2. Core Development Phase (May 2020 – March 2024)

Jems was developed iteratively using the **Agile Scrum methodology**, allowing for continuous testing and feedback integration. The development of Jems followed key milestones, which were defined along the development of specific modules in Jems.

- **Project lifecycle up to decision** – Initial functional release enabling programme setup, application forms, and decision-making workflows.

- **Contracting to payments** – Introduction of modules allowing for contracting of projects, reporting, management verifications and payment functionalities.

- **Reporting to EC, audit and closure** – Full implementation of the payment module including payment applications to EC and accounts, as well as the audit and corrections module and project closure.

By April 2024, Jems had completed its 10th release, delivering all essential functionalities required for Interreg programme management.

2.2.3. Additional Development Phase (April 2024 – December 2024)

The Additional Development Phase of Jems focused on implementing improvements based on direct input from the user community. Through structured voting rounds, programmes were able to prioritise the most valuable features, fostering a strong sense of ownership and ensuring alignment with real operational needs. This participatory approach led to the successful development of enhancements such as bulk file downloads, improved data export options, and expanded verification tools.

The release of Jems version 12 in December 2024 marked the completion of the additional development phase and the conclusion of active system development.

2.2.4. Maintenance phase (2025 – 2029)

During the Maintenance Phase of Jems, Interact continues to provide core support, including bug fixes, user assistance through the helpdesk, and minor enhancements where feasible. This phase focuses on ensuring system stability, addressing technical issues, and adapting the software to evolving regulatory and operational needs.

A key challenge in this phase is ensuring that sufficient resources in Interact as well as Cloudflight remain available to respond to future legislative changes or security requirements without reinitiating large-scale development.

Feedback:

While the definition phase allowed to build up a good basis for the project implementation, it felt that too much time was spent on developing theoretical concepts, while the start of the actual development work had to be postponed until a development team could be set up at the side of the software development company.

During the core development phase, the development of features was driven by the timeline of the programmes. Surveys provided valuable insights into how programmes differ regarding requirements and also with regard to their timelines. Early adopted programmes defined start dates for calls putting pressure on early delivery of certain features.

During the additional development phase, voting was found as a good practice. It generated a high level of participation among stakeholders and fostered a sense of ownership within the community. The process was particularly effective in helping to prioritise new features and provided a clear justification when specific requests had to be turned down. For Interact, voting offered reassurance that the features selected for implementation were supported by the broader user community. While there were occasional interpretation challenges - since each feature had to be summarised in just a few lines - these were manageable. Once a feature reached the top of the voting list, any remaining ambiguities were clarified in close cooperation with the Core Group, which consistently worked smoothly.

The stakeholders appreciate the Jems support guaranteed until end of 2029, however, the lack of additional budget for development of extra features as well as the unsecure future after 2029 remain an open discussion.

2.3. Definition of the scope

The scope of Jems was defined through a structured identification of use cases, which provided the foundation for planning and prioritising the system’s functionalities. Early in the project, Interact compiled an initial list of anticipated use cases covering the entire programme and project management lifecycle, from programme and call setup to final project closure.

This initial definition was based not only on Interact’s expertise but also on broad stakeholder input: following a recommendation from the feasibility study conducted by Ernst & Young, a stock-taking exercise was carried out. In May 2020, a dedicated survey among all Interreg programmes collected information about technical requirements, the use of optional functionalities, and desired improvements for a future monitoring system. The results of this survey directly fed into the development of the initial use case catalogue. Additionally, the project setup of Jems strongly built on the experience with the existing eMS system to ensure an optimal use of the knowledge and lessons gathered from the 2014–2020 period.

In parallel, the Jems Core Group, composed of representatives from several Interreg programmes, was actively involved in refining and validating the use cases to ensure they matched practical programme realities.

Each use case was categorised using the "Must," "Should," and "Could" prioritisation logic:

- **Must haves**: Essential features that had to be addressed first;
- **Should and Could haves:** Features to be developed depending on resource availability and evolving user needs, with "Could" features potentially being dropped if necessary.

The use case catalogue was continuously updated throughout the development process, allowing Jems to remain flexible while maintaining a clear and coherent scope. Although pressure from programmes led to some "beyond must" features being advanced earlier than foreseen, the use case-based approach ensured that Jems evolved systematically, balancing regulatory obligations, technical feasibility, and stakeholder expectations.

Feedback:

Feedback gathered from the Core Group and project documentation indicated that although the identification of use cases and the classification into "Must," "Should," and "Could" categories provided a clear foundation for planning, the communication around the process was sometimes difficult. Interact had initially communicated that only "Must" use cases would be developed first; however, strong early pressure from programmes to go beyond a minimum viable product led to the implementation of some "beyond must" features sooner than planned. This adjustment created resource strains and impacted the categorisation and scheduling of "Should" and "Could" features, especially during the first year of development. Moreover, there was an overestimation of the development speed and capacity of the IT provider, which affected the early levelling of priorities.

Despite these difficulties, the approach allowed the system to remain flexible and adapt to evolving user needs, and the involvement of the Core Group in defining and refining the use cases was seen as a critical factor in aligning the system with programme expectations.

2.4. Agile as a methodology

Agile, specifically the Scrum framework, was selected as the development methodology for Jems to address the particular challenges and requirements of the project.

Three main factors drove this choice:

- **First, the extremely tight timelines made Agile essential.** Early adopted programmes required a functioning monitoring system with core functionalities available as early as Spring 2021. Agile made it possible to release software iteratively and quickly, allowing core functionalities to be delivered in phases without waiting for the entire system to be completed. This flexibility was critical in keeping up with the operational needs of programmes preparing to launch their first calls.

- **Second, Agile allowed Jems development to begin without final requirements.** When the development phase started, the EU regulatory framework for 2021–2027 was still under negotiation. Agile's flexibility meant that initial development could proceed based on draft assumptions, with rapid adjustments made as soon as legal and operational requirements became clearer. This avoided costly delays and ensured that Jems remained aligned with the evolving legal context.

- **Third, the values of the Agile Manifesto – such as stakeholder collaboration, transparency, and responsiveness – were crucial** and directly addressed lessons learned from the previous eMS project. In contrast to the more closed development process of eMS, Jems actively involved programmes through bi-weekly sprint reviews, direct feedback channels, and prioritisation exercises like the user-driven voting rounds. This continuous engagement helped build strong ownership among users and ensured that the system closely matched the real needs of Interreg programmes.

In summary, Agile provided the necessary speed, flexibility, transparency, and community involvement to successfully deliver a complex, shared monitoring system under challenging and changing conditions.

Feedback:

While the Agile methodology brought significant benefits to the Jems development, it also posed challenges for the stakeholder community, particularly at the beginning. For many programmes, Agile was a completely new approach and they struggled with the dynamic nature of it, as they were accustomed to having detailed technical specifications and clear delivery schedules far in advance. Agile's iterative process, which only defined functionalities in detail shortly before their development, led to occasional frustrations.

Some programmes were also required to define their procedures based on functionalities that were still under development. Sprint Reviews and early demonstrations helped mitigate these issues by offering early visibility and feedback opportunities. However, communicating the overall project picture remained difficult at times, and the lack of a fixed long-term plan caused some uncertainty.

Despite these challenges, stakeholders appreciated the transparency, regular interaction, and ability to influence developments through feedback loops and voting rounds. Over time, Agile strengthened ownership and trust within the community.

Lessons learned from the experience suggest that, in future projects, additional initial training on Agile roles and processes for all stakeholder groups, combined with more structured early-stage planning, could further enhance the effectiveness and acceptance of Agile in large-scale, community-driven developments like Jems.

2.5. Selecting a service provider

To ensure the rapid initiation of the Jems development process, Interact leveraged the Bundesbeschaffung GmbH (BBG) procurement framework instead of conducting a separate Europe-wide tender. The decision to use BBG was primarily driven by time constraints, as the Jems development needed to start immediately after approval to meet the critical deadlines for programme implementation.

BBG is Austria’s central public procurement agency, which pre-negotiates contracts with IT service providers through comprehensive Europe-wide procurement procedures. This approach allowed Interact to directly access pre-vetted IT suppliers, significantly reducing the administrative burden and accelerating the contracting process.

Key Advantages of BBG Procurement in Jems:

- **Time Efficiency** – Instead of spending months on a standalone tendering process, Interact was able to immediately select a provider from BBG’s framework. This ensured that the first development phase began in 2020, aligning with the urgent timeline.
- **Pre-Qualified Suppliers** – The BBG framework guarantees that all listed IT companies meet high technical and legal standards, ensuring that the selected provider, Cloudflight, was capable of delivering a complex IT system like Jems.
- **Reduced Procurement Risks** – BBG conducts thorough evaluations of suppliers, ensuring compliance with EU procurement rules. This minimized legal risks for Interact and eliminated uncertainties associated with conducting an own procurement process.
- **Cost-Effectiveness** – The BBG framework bundles demand across multiple public sector entities, leading to competitive pricing and better contract conditions compared to individual tenders.
- **Flexibility in Contracting** – Using BBG allowed for a more adaptive contract structure, enabling iterative development under an agile methodology, where requirements evolved based on stakeholder feedback.

Feedback:

While the BBG framework was critical in ensuring a fast start for Jems, it also introduced some constraints. In 2022, an update to the BBG framework required Interact to commit to ordering development roles until the end of the maintenance period. Otherwise, a change of development company mid-project could have been necessary, potentially causing delays and instability. Despite this, the time saved, legal compliance, and cost efficiency far outweighed the risks, making BBG procurement a key factor in Jems' success.

3. Development process

The development process refers to a structured set of phases applied in the creation of Jems. The adoption of an agile development methodology marked a significant shift compared to the approach used for eMS. Several key changes were introduced based on lessons learned from the previous system. One of the main criticisms of eMS was its limited focus on user friendliness. In response, Jems placed greater emphasis on user experience (UX), dedicating additional development capacity to improve interface design and usability. User involvement was strengthened through targeted user testing to ensure that the system was intuitive and met the needs of different user groups.

Given the importance and added value of a common monitoring system for programmes, Interact was advised to put additional resources into its development and maintenance. In order to estimate the required resources for Jems, a feasibility study was conducted by Ernest & Young. Based on the recommendations from the feasibility study, the resource allocation for the Jems project has been substantially increased compared to its predecessor, eMS, ensuring that the staff allocation and budget were proportional to the project's impact and financial risk.

3.1. Team dynamics, team size and collaboration

This section describes the evolution of the development team, the level of collaboration between Interact and Cloudflight, and lessons learned regarding the optimal team size for a project of this complexity and scale.

3.1.1. The team

At peak development, the Jems project team included:

Interact

- 1 Project Manager (not directly involved in Jems development)
- 1 Product Owner (PO) - also partly taking over the role of Business Analysts
- 2 Business Analysts (BAs) (i.e. Interreg requirements analysts)
- 2 IT Managers - also partly taking over the role of Business Analysts, especially for technical requirements and feature specification)

Cloudflight

- 1 Project Manager
- 1 UX designer
- 1 Scrum Master / 1 Quality Assurance (QA) manager
- 1 Architect – also partly working as a developer
- 7 Developers

During the second year of development, some adaptations occurred within the team structure. The dedicated UX designer position was discontinued, with UX responsibilities integrated into the regular feature refinement and development workflow. Quality Assurance (QA) responsibilities initially managed by the Interact IT Managers were transferred to Cloudflight, following the departure of one IT Manager at Interact. The new QA Manager position at Cloudflight also assumed the responsibilities of the Scrum Master, resulting in the dedicated Scrum Master role being fully phased out during the final years of development.

Although some personnel changes occurred - including one change of Architect and two changes of Scrum Master on the Cloudflight side - these transitions were managed smoothly and did not disrupt the continuity or stability of the development process. The team's ability to absorb changes without significant delays is a reflection of the strong collaboration culture built between Interact and Cloudflight.

Feedback:

The team setup successfully covered the knowledge and skills required to implement a project of this complexity. Collaboration between Cloudflight and Interact was professional and developed into a daily working reality, with both organisations operating, to a certain extent, as one integrated team. The high level of involvement by Interact in the development process was somewhat unusual for Cloudflight, where typically the roles of Business Analysts and Product Owners are filled internally. By maintaining close collaboration with all stakeholders, drafting user stories, actively participating in refinement meetings, and steering sprint planning from our side, Interact ensured that the system’s development evolved in line with programme needs. Ultimately, the strong partnership between Interact and Cloudflight contributed significantly to the success of the project.

3.1.2. Team Dynamics and Collaboration

From the beginning, building a close and transparent collaboration between Interact and Cloudflight was prioritised. The teams worked together intensively through regular planning sessions, daily communication channels, and joint refinement, testing, and retrospective meetings. Despite being geographically dispersed and working remotely for most of the project duration, the team managed to maintain a strong sense of unity. Regular sprint ceremonies, thematic meetings, and internal synchronisation meetings helped bridge communication gaps, while early in-person gatherings, such as the team kick-off in Vienna, contributed significantly to establishing trust and personal bonds.

A major strength of the collaboration was the open and horizontal communication culture. Input from all team members was valued regardless of seniority, creating an inclusive environment that motivated the team and maintained high engagement over the years. The core team composition remained relatively stable, despite some changes in individual roles, which helped sustain trust and efficiency across the project's lifecycle.

Nevertheless, challenges were encountered, particularly when the team size grew during peak development periods. A larger team increased the complexity of communication and decision-making, sometimes causing meetings to lose focus and prolong discussions. Clear role definitions, structured workflows such as pre-refinements and tech-checks, and a strong focus on iterative retrospectives helped to mitigate these challenges. However, it became evident that for future projects, smaller teams or a division into two more focused teams could better balance development capacity and communication efficiency.

One of the key lessons learned was that, for a project of this type, a core team of 5-7 developers, complemented by 2–3 Business Analysts and strong coordination roles (Product Owner, Project Manager), would likely represent an optimal balance between development capacity and communication efficiency. While a larger team accelerated feature delivery, it also increased the need for structured processes to prevent bottlenecks and maintain a common understanding of priorities and specifications. Early and comprehensive training in agile practices for all team members would further enhance project efficiency and collaboration.

Overall, the Jems team succeeded in building an environment that enabled complex software development across organisational boundaries, demonstrating that close collaboration, shared ownership, and mutual respect are essential ingredients for success in cross-institutional IT projects.

Feedback:

The collaborative atmosphere between Interact and Cloudflight was seen as a key factor for success, characterised by openness, trust, and shared responsibility. The initial role of the Scrum Master was especially important for setting up agile processes, facilitating retrospectives, and ensuring positive team dynamics. Given Interact's limited prior experience with agile project management, this support was crucial during the early phases of the project.

However, as the team matured and became increasingly self-organised, the added value of a dedicated Scrum Master was no longer clearly visible. In the later years of development, the role was fully phased out. Likewise, certain agile ceremonies, such as bi-weekly retrospectives, were no longer held on a fixed schedule but were organised when necessary, reflecting the team’s growing autonomy and a natural fatigue with overly formal procedures.

Regarding team size, during peak development the team grew quite large. While having a larger team sometimes helped to accelerate the delivery of critical features by allowing multiple scenarios to be explored, it also increased communication overhead and introduced many opinions, which at times slowed down development speed. A division into two smaller teams was considered but ultimately not implemented. With good communication flows and a clear division of responsibilities, such an approach could have further improved efficiency without compromising quality.

For future projects of similar complexity, it would be beneficial to invest early in strengthening agile and project management knowledge on the client side. Building a solid internal understanding of agile methods and project management tools before engaging with the development company would have enabled more proactive engagement in project steering, and allow the Interact team to challenge and shape project processes more confidently, ensuring that the development setup is fully tailored to the specific needs of the organisation.

3.2. Organisation of the development process

The development of Jems was organised following an agile methodology, applying the Scrum framework. Agile was selected to enable flexibility, manage evolving requirements, and respond rapidly to unforeseen challenges during the development. This methodology supported short, iterative development cycles, while also requiring a highly structured process involving continuous prioritisation, sprint-based work, frequent stakeholder engagement, and regular feedback loops. The agile approach further incorporated regular retrospectives to improve both the product and the development process itself. The following sections outline the organisation of the development steps, the meetings introduced, and the tools used.

3.2.1. Development Steps, Meetings, and Tools

The development of new features followed a structured and collaborative process involving multiple steps ensured continuous alignment between the development team and stakeholders.

While core agile ceremonies like Refinement meetings, Sprint Planning, Sprint Reviews, and Sprint Retrospectives were integral to the organisation of the development process from the beginning, additional steps in the workflow evolved over time based on the feedback given by the project team (Interact team and development team) in the retrospectives.

Process for feature development

Initially, Interact gathered requirements through consultations with the Jems Core Group, thematic groups, the User Group, and directly with specific programmes that had particular needs.

Based on these requirements gathered, **Business Analysts (BAs) drafted user stories** in the project management tool Jira. Each draft story was then consulted with a developer to ensure technical feasibility. Before the story could move forward, a **pre-refinement meeting** was held among Interact staff (excluding the project manager) and the project manager from Cloudflight to verify that all aspects of the requirements had been captured and the specifications in the user story were sufficiently clear and precise. If gaps were identified, the story was sent back for revision; otherwise, it proceeded to a **technical check**, where a developer assessed the technical feasibility and discussed potential implementation details with the BA. This check could either result in a technical validation or identify issues that would require another loop potentially including the reconsideration of business requirements and further specification by the BAs as well as all consecutive steps to reach technical validation.

Following technical validation, the story moved to **effort estimation in refinement meetings**, where developers used the **Fibonacci estimation method** to assign story points. If a story was unclear or too large, it was either sent back for clarification to the Bas or split into smaller, more manageable parts. Once the story was ready, the Product Owner (PO) assigned it to a sprint for implementation.

At the beginning of the sprint, a **kick-off meeting** took place between the BA and the assigned developer(s) to clarify the tasks. Depending on the complexity, the work could be divided among several developers. After implementation, developers performed a **quick test** together with the BA to check the basic functionality. If the initial test was successful, the story proceeded to **code review**, where it required two independent approvals ("two thumbs up") from other developers, ensuring code quality.

Once reviewed, the story was deployed to a Jems test environment. Developers conducted **technical tests**, after which the story was moved to the Test-by-PO stage. Interact (PO or BAs) then **tested the feature against business requirements** and, if all tests passed, marked the story as done.

UX-related features

For UX-related features, at the beginning of the project a dedicated UX designer was involved and specified UX concepts and templates. Early UX concepts often did not match real user workflows and were, in many cases, not implemented. While UX templates were developed to provide a standardised design approach, the variety of programme requirements led to many exceptions, resulting in some inconsistencies.

After that initial stage, UX tasks were integrated in the tasks of the Cloudflight project manager and UX features followed a similar process as any regular features. The Product Owner decided in consultation with the UX project manager on the prioritisation and timing of UX stories.

Bug handling process

The procedure for managing bugs differed from that for feature development. Bugs could be reported by Cloudflight developers, Interact colleagues, or programme users via the Jems Helpdesk. Suspected bugs were logged in a Bug Triage epic in Jira.

Weekly Bug Triage meetings were conducted by the Cloudflight QA manager and an Interact BA to assess reported issues. Depending on the findings:

- Confirmed bugs were assigned to a bug epic for a future release or classified as low-priority known issues.
- Issues found to be related to business logic were clarified with the reporter.
- Some issues were reclassified as UX improvements and forwarded accordingly.

Confirmed bugs were categorised by severity:

- Highest priority: addressed immediately, potentially interrupting other development activities.
- High priority: to be addressed as high priority once a developer was available to start a new story and to be finished within the sprint.
- Medium priority: resolved before the next release.
- Low priority: resolved opportunistically during sprints.

Bugs were considered sprint overhead and not subject to separate estimation. Serious bugs requiring immediate correction resulted in **Hotfix releases**, and automation tests were added where feasible to prevent recurrence.

Tools utilised

The development process was supported by several key tools:

- **Jira:** for project management, sprint planning, issue tracking, and reporting
- **Confluence**: for collaborative documentation, specifications, and sprint records.

- **Test Environments**: for early verification of completed features.
- **Jems Helpdesk**: for user support, bug reporting, and feedback management.

Feedback:

The intensive structure of the Jems development process, including the large number of meetings and workflow steps, was generally well received by the project team at Interact and Cloudflight side, albeit with some critical reflections. High levels of communication and availability from both Interact and Cloudflight were widely acknowledged as essential factors for the project's success. However, feedback also indicated that the frequency and intensity of meetings at times felt exhausting. Team members suggested that greater discipline in time management, clearer agendas, and stricter focus during meetings would have improved efficiency and reduced the feeling of meeting overload.

The decision to introduce multiple preparatory steps before feature development - such as pre-refinement meetings, technical checks, and structured quick testing - was positively noted, particularly given the size and complexity of the team and the stories handled. While some team members observed that a smaller team might have allowed for a more streamlined process with fewer formal steps, it was generally agreed that the structured workflow helped maintain the high quality and consistency of the final product.

Sprint Reviews, held every two weeks, served as a key mechanism for presenting newly developed features and UX improvements to stakeholders. These sessions consistently recorded high attendance across the entire project lifetime, although it was noted that active verbal feedback often came from a limited group of participants. Feedback collected during these reviews was immediately incorporated into the Jira backlog, either as refinements to existing user stories, new bug reports, UX improvements or change requests.

In terms of development tools, the project team confirmed that Jira, together with its extensions such as Confluence, provided a solid foundation for managing the project’s complexity. Jira's versatile functionalities - including sprint boards, backlog management, timeline views, and reporting - supported transparent coordination and monitoring of progress. Although certain aspects, such as retrieving archived epics or managing complex test structures, required more in-depth knowledge of the tool, Jira’s central role was unanimously considered critical to the success of a project of this scale. The decision to integrate the Jems Helpdesk into the Jira environment and to maintain documentation and the user manual within Confluence was also regarded as highly beneficial, as it improved accessibility, cross-referencing, and operational consistency.

Overall, while some process-related improvements were identified, the Core Group and wider stakeholders generally expressed high satisfaction with the organisation of the development process, the degree of transparency, and the responsiveness of the project team.

3.2.2. Prioritisation

Prioritisation in the Jems development process was a central activity, ensuring that development work was aligned with programme needs and available resources. The process evolved over time and differed slightly depending on the type of item to be prioritised, combining structured stakeholder involvement with agile responsiveness to changing needs.

Prioritisation of Features

At the strategic level, the Jems Core Group participated in prioritising major functionalities at the epic level - larger feature blocks representing significant modules of the system such as project application, contracting, and reporting. The definition of the epics was based on the use cases identified at the project start and recurrently updated also considering the categorisation of the use cases according to the "Must," "Should," and "Could" logic. The collaborative approach between Interact and the Core Group ensured that development focused on delivering core functionalities critical for programme implementation across the Interreg community at the right time and in the right quality.

At the operational level, feature prioritisation occurred bi-weekly during sprint planning meetings, where the PO assigned specific user stories to sprints based on their importance, technical readiness, and any identified dependencies. This cycle provided flexibility to adapt priorities to newly emerging needs without losing sight of the strategic roadmap.

During the Additional Development Phase (April–December 2024), prioritisation was further expanded through the introduction of a Wishlist mechanism. Programmes were invited to propose additional feature requests, and a democratic voting process determined which developments would be prioritised for implementation. This approach significantly strengthened stakeholder ownership and ensured that the highest-priority enhancements implemented in this phase were both needed and community-endorsed.

Prioritisation of UX Improvements

In the initial stages of the project, UX improvements were prioritised separately. The UX designer at Cloudflight, working closely with the Product Owner, defined priorities for user experience tasks, operating in parallel with general feature development. Following organisational changes and the departure of the dedicated UX designer, UX prioritisation was fully integrated into the broader feature prioritisation workflow. UX tasks were then treated as any other user story within sprint planning, ensuring continuous attention to user experience without requiring a separate management track.

Prioritisation of Quality Assurance (QA) Issues and Bugs

Initially, bugs and QA-related issues were handled as part of the general sprint overhead, addressed as they were discovered during development and testing. However, as the system matured and the volume of bug reports increased, a more formal approach was adopted. A weekly Bug Triage meeting was introduced, led jointly by the QA manager at Cloudflight and a Business Analyst at Interact. During these meetings, newly reported bugs were assessed and prioritised according to their severity into highest, high, medium or low priority bugs (see above). High and critical priority bugs were scheduled for immediate or sprint-based resolution, while lower priority issues were addressed when capacity allowed. This approach helped ensure that critical issues were dealt with efficiently without disrupting overall development progress, complementing the broader agile workflow.

Feedback:

In terms of feature prioritisation, stakeholders sometimes had different interpretations of priorities. One of the main challenges identified by the Core Group was that strategic prioritisation was carried out at the epic level, based on the use cases and their classification into "Must," "Should," and "Could."

However, Core Group members were neither involved in the detailed specification of individual features and technical refinement discussions with developers, nor in the operational prioritisation at the level of the sprint planning. As a result, it was sometimes difficult for stakeholders to fully understand why, despite following the agreed strategic prioritisation by the Core Group, certain functionalities were specified differently than expected or why some features and functionalities were prioritised ahead of others at the operational level.

The introduction of a democratic voting process during the Additional Development Phase was regarded as a best practice. It brought greater transparency, strengthened ownership among programmes, and provided a clear mandate for decision-making. Although the voting process required more effort and coordination, it was seen as essential for achieving broad acceptance and could have been beneficial if implemented even earlier in the project.

Integrating UX prioritisation into the overall feature planning improved relevance, and user tests proved valuable in validating design choices across different stakeholder profiles.

In the area of Quality Assurance (QA) and bug handling, stakeholders acknowledged the strong commitment to quality as a major strength. Structured bug triage meetings, combined with automated, end-to-end, and manual testing, helped maintain a high level of reliability throughout the development process. Although this focus on quality occasionally slowed down development speed, it was recognised as essential for securing the long-term sustainability and robustness of the system.

Overall, the prioritisation approach balanced structured stakeholder involvement with agile flexibility. Clearer communication of prioritisation logic and adapting agile principles to include longer-term planning where necessary were seen as critical factors for building trust and ensuring that Jems evolved according to the real needs of the Interreg community.

4. Cooperation with Stakeholders

When setting up the Jems project, the intensive involvement of stakeholders was one of the main goals. Based on the lessons learned from eMS, the project management structure with a small Core Group steering the project together with Interact and an extended User Group as observers with the possibility to provide input had turned out to be effective and was therefore taken up for the Jems project.

Some voices in the programmes suggested that the development of a new monitoring system should be more top-down to increase efficiency. However, this should not deflect from the fact that a central success factor of eMS was the participatory and inclusive approach to involving programmes, as they are the ones ultimately using the system. Therefore, communication was identified as key for the success. A recommendation taken up from the eMS impact evaluation was to make more resources available for the communication with the stakeholders.

4.1. Communication tools

In order to allow for an effective and efficient communication with stakeholders, several communication channels were established.

4.1.1. Online Communities

Already from the beginning of the project, Interact has established Jems online communities. The Communities are the backbone of Interact's communication with stakeholders and ensured an efficient and targeted communication flow in both directions.

- **Thematic Network on Jems:** The main community is called the "Thematic Network on Jems" and anyone interested to follow the Jems development was able to join this community. In this community regular news about the project together with summaries of all relevant meetings were published. This community was used to announce any relevant events for the user community. At the same time, this community provided a platform for users to exchange on certain topics and discuss about specific issues with other users.

- **Sub-communities:** The main Jems community has three sub-communities:

    - **The "Jems User Group"** is the community of all license-holders and is the main communication channel of Interact to the Interreg Programmes using Jems as their electronic monitoring system. In this community, Interact informs all users about releases, critical bugs and hotfixes. The information provided via this community is mainly product related.

    - **The "Jems Core Group"** is the community of all Core Group members, where Core Group meetings, test days and thematic group meetings were announced and relevant documents were stored.

    - **The "IT Group"** is the group of IT specialists from the Core Group.

4.1.2. Other communication channels

Additionally, Interact established several other communication channels.

- Interact portal – a one-stop-shop for all relevant information about Jems.
- Posts on social media to inform the wider public about latest news.
- Sprint review meetings – bi-weekly meetings to informed interested parties about latest developments also allowing them to provide direct feedback.
- Online events – especially during the first years of development, Interact invited all stakeholders to join online milestone events to inform about the progress of the project.
- Helpdesk – an online platform allowing users to pose questions, transmit suggestions, and report on bugs in a structured way.
- Jems e-mail – one e-mail to reach all Interact staff working on Jems.

Feedback:

In the two License Holder Surveys in spring 2023 and in spring 2024, stakeholders were asked to rate their experience with communication channels established by Interact. In 2023, 90% of the programmes rated their overall experience as very positive or rather positive. Even though only 4 programmes rated their experience as rather negative, Interact has further levelled up the communication efforts and in the License Holder Survey 2024, about 98% of all programmes were very satisfied or satisfied with the communication channels established by Interact.

4.2. Jems Core Group

Together with Interact, the Jems Core Group had a crucial role in the Jems development.

4.2.1. Establishment of the Core Group

The establishment of a Jems Core Group was considered a crucial factor in ensuring the success of the project. The main goal of establishing a Core Group was to bring together a smaller group of interested and dedicated Interreg Programmes supporting the development of Jems on behalf of the whole Interreg community.

In order to ensure the establishment of all project structures in due time, already in 2019, Interreg Programmes had the possibility to apply for the participation in the Jems Core Group. The selection criteria were that Programmes had already gained some experience with developing eMS or their own monitoring system as well as the commitment to use Jems and to provide sufficient staff resources to support all the Core Group activities. Interact also made sure that different Interreg strands are represented in the Core Group, such that members can bring in their knowledge about the specific requirements ensuring the software meets the needs of the bigger Interreg community.

Overall, eight Interreg Programmes were selected to become members of the Jems Core Group, representing four Programmes who had been members of the eMS Core Group and four additional Programmes, who had partly been eMS users and partly had their own monitoring system.

Feedback:

The Core Group members considered it a good practice to select the four programmes who had been members of the eMS Core Group into the Jems Core Group so that the expertise and experience from the previous period was not lost.

4.2.2. Role of the Core Group

The Core Group was involved in different tasks:

- **Definition of requirements:** From the beginning, the Core Group was involved in the definition of the scope of the software by defining the list of use cases that had to be covered in Jems. Throughout the implementation phase, the Core Group defined requirements for Jems modules and features within thematic group meetings. Within those workshops, the participating Core Group members provided knowledge about the requirements and their workflows and made decisions on specific options for implementing certain features. Based on the input provided and the decisions made by the Core Group, the Business analyst of the Interact team defined the exact feature specifications in the user stories.

- **Prioritisation of development work:** The Core Group was involved in the decision making concerning the prioritisation of certain features and modules. While the upper level of the milestones defined in the project plan, the lower level of prioritisation was done by the Product owner of the Interact team (consulting also Business analysts) together with the Core Group. This prioritisation of the Core Group was done at the level of epics (bigger features or modules) for every release, but not at the level of single user stories – always within the limits of development capacity and effort estimated for epics as defined by the Interact team together with the development company. While the Core Group was involved in the release planning defining the scope of development work over a period of several months, the Interact team decided the organisation of work with the software development company within the single sprints.

- **Testing of features:** The Core Group was encouraged to continuously test the features delivered. Before the releases, Interact organised a joint test day with the Core Group members setting up a test plan for ensuring in this way that Interreg Programmes were involved in the testing of all features and modules before they were delivered to the license holders.

- **Ambassador toward the User Group:** The Core Group held an ambassador function towards the User Group, having deeper insight into the development process and being involved in the decision-making process.

Feedback:

The Core Group and its members actively took up the tasks they had committed for. The persons assigned to the Core Group brought in a lot of knowledge and expertise into the project. There was a very positive working atmosphere from the beginning allowing for a good cooperation based on mutual trust. Based on the legacy from eMS and the new setup of the project structure, the separation of roles and tasks between the Interact team and the Cour Group was a point of discussion throughout the first years of implementation. The Core Group were requesting deeper involvement in the actual development process by writing and reviewing user stories, being involved in the estimation of effort and in the code review and acceptance process. Based on the lessons learned from eMS, the allocation of sufficient staff resources within Interact allowed to cover such tasks within the project team and the involvement of stakeholders in the daily organisation of development work and decision making would not have allowed an efficient and effective implementation of the software. Therefore, the role of the Core Group was limited to the higher-level roadmap planning (prioritisation of epics, releases etc.), while the detailed planning and decision making concerning the commitment of work to the development company (writing and estimating user stories, planning of sprints and acceptance of development work) was done by the Interact team. This separation of tasks was finally defined and agreed in 2022 in the Jems Core Group Cooperation Principles.

Core Group members took their role as ambassadors towards the User Group serious. They were backing up decisions made concerning the higher-level roadmap and release planning. At the same time, they provided peer-support to other, less experienced programmes. In some cases, even local networks with other programmes of the same Member States were established to provide support and collect requirements.

Even though Core Group members had their own specific needs and programme interests in certain features and developments, the representatives tried to make decisions in the interest of the overall community.

4.3. Jems User Group

The User Group represents the pool of Interreg programmes which have signed a Jems license agreement or consider to use Jems as their monitoring system. For many of the Interreg Programmes using eMS as a monitoring system in the 2014-2020 period, it was clear that they would sign a license agreement for Jems. Over time, also many additional Interreg Programmes joined Jems. The first license agreement was signed beginning of 2021. Overall, more than 50% of all Interreg Programmes, namely 48 programmes, are using Jems as their monitoring system for the 2021-2027 period.

While the User Group is supported by Interact in their use of Jems, it is also considered a network of users providing mutual support to each other and is an interface to representatives of particular types of Jems users (e.g. controllers, beneficiaries), in order to investigate requirements and to perform user tests with them.

While in the past, the eMS User Group was mainly involved through User Group meetings and through an online community platform, the bi-weekly sprint review meetings provided a new level of transparency with regards to the latest developments. This even led to follow-up meetings self-organised by the users without the help of Interact, e.g. post-sprint review meetings, but feeding back their discussion results into the development process.

Through various communication channels the User Group was able to submit requirements and ideas for improvement. The immense feedback provided in the sprint review meetings and via the helpdesk helped the Interact team to meet the requirements, improve the quality and ensure that the software is functional and usable. At the same time, many wishes were not considered as must-haves and could not be taken up during the core development phase, which sometimes created frustrations, as expectations could not be met. For some Interreg Programmes, which had not used a community monitoring system before, the adaptation to a harmonised workflow was challenging. Also underlying concepts, like the HIT templates, were not fully accepted by all Jems users, as they wanted to go their own way.

As a new model, Interact established a wishlist voting system during the additional development phase, which gave the license holders the ability to decide about the further improvements and functionalities implemented in the last nine months of the development phase. Based on the results of two voting rounds with about 190 wishes from the User Group put up for voting, Interact was able to implement more than 35 additional features and export plugins bringing the highest added value for the users.

Feedback:

While users welcomed the higher transparency and possibility of continuous involvement through the Agile Scrum working methodology, this also caused certain uncertainties among the users due to a lack of detailed specifications for upcoming modules and features. While the high-level road map concerning the overall scope of the software and the milestones was provided to the users, the exact timeline and scope of the single releases was only decided case-by-case and release dates including the list of features were usually published only few weeks or months before the actual release. Based on this feedback, Interact tried to level up communication efforts to reduce uncertainties and doubts. Still, there were critical voices and the management of expectations from Interact side had to improved, proactively addressing the criticism or frustrations of certain programmes with bilateral meetings and specific support activities.

The new concept of wishlist voting was very well received by the User Group, which also became visible in the high participation rate of programmes in both voting rounds. It therefore is considered a good practice that can ensure broad acceptance and stronger stakeholder commitment.

4.4. Other stakeholder groups

Besides the Jems User Group and Core Group, there were also other stakeholders involved in the Jems project.

Among the internal stakeholders, there was the Interact Monitoring Committee (MC), which is the body approving the project in 2019. Throughout the first years, the MC was skeptical about Interact's ability to implement the project within the given timeframe and budget. This is why the MC established a specific Steering Group which should follow-up regularly on the process of Jems. Interact had to provide regular reports on the progress of the project and only in 2023, based on the evidence provided by the Interact team, the MC gained sufficient confidence that the product could be delivered on time and within the given budget.

Besides the MC representing the Member States, also the European Commission (EC) was a stakeholder in the Jems project. The direct involvement of the EC was very limited. For certain modules, Interact was seeking guidance or interpretation of rules from the colleagues of the EC. At the same time, Interact offered information about Jems to the EC, e.g. by organizing a Jems training for Interreg desk officers, which had a very high participation.

As external stakeholders, also the IT service providers of the Interreg Programmes using Jems were involved in the project. Besides hosting the application, external IT providers also did their own developments of smaller features and plugins on top of the Jems standard package. Interact invited IT service providers to participate in the Jems User Group meetings and in the Sprint Review meetings as well as in the helpdesk.

Feedback:

The cooperation with the European Commission was very fruitful. The feedback and guidance provided concerning e.g. the reporting requirements in SFC or the submission of the population for the common sample was very helpful for the Interact team to correctly define the features in Jems. Also, the cooperation with the Monitoring Committee was positive, even though there were certain critical voices. The close monitoring of the progress and spending helped the Interact team to ensure that the project was right on track and could be delivered within the given timeframe and budget.

The cooperation with external IT providers proofed to be fruitful. IT providers gave positive feedback concerning the quality of the system and the code, so it was easy for them to build own developments on top. Also, the direct involvement of IT providers in the User Group meetings brought positive learning effects, while at the same time there were voices stating that those places should have rather been taken by Programme representatives. Some IT providers caused frictions between Interreg Programmes and Interact, when they blamed the software package for certain miss-functionalities or missing features, while in reality they had missed to install certain updated or hotfixes for their clients.

4.5. Other related projects

Besides other stakeholder groups, there were also other related projects in Interact, which had an impact on Jems. As an electronic monitoring system, Jems stands in the center of several integrated Interact tools for the management of Interreg Programmes and their projects. HIT – the harmonized implementation tools – builds with its templates covering the whole project lifecycle the basis for Jems. At the same time, Jems feeds into the data sharing, analysis and visualization platforms developed by Interact, namely INDEX and keep.eu.

Feedback:

The parallel implementation of several Interact tools required a high coordination and collaboration effort between Interact colleagues to ensure consistency and interoperability between all the tools. This was mainly possible by the mutual involvement of all colleagues in the Core Group meetings of the concerned tools.

Even though it was clear that Jems could not have been built without having the harmonized templates of HIT as a basis, the Jems Core Group and User Group did not always feel a clear ownership of the HIT templates and there were many requests for deviating from the templates. Additionally, the HIT process did not look into processes and, therefore, the HIT tools required certain adaptations when implemented in Jems. Certain information could have been provided in a more useful and cost-efficient way, if not strictly following the templates designed by HIT.

5. Jems as a product

Stakeholders had high expectations for Jems as a product, anticipating a significantly improved and more user-friendly successor to eMS. Beyond functional continuity, there was a strong demand for enhanced usability, greater technical flexibility, and a system architecture capable of supporting long-term sustainability and integration with evolving digital ecosystems.

5.1. Technical aspects

Jems follows a classic modular web application architecture, carefully designed to deliver high system performance, maintainability, and interoperability with external systems. Architecture decisions and strict coding standards were fundamental to meeting critical requirements identified from lessons learned with the previous eMS system.

5.1.1. System Architecture

The Jems system is composed of the following core components:

- **Spring Boot application (Kotlin):** The backend server application is developed in Kotlin, using the Spring Boot framework. It exposes a RESTful API that handles communication between the frontend and backend systems.

- **Angular frontend:** The user interface is built with Angular, offering a responsive and modern user experience.

- **MariaDB:** A relational database system that stores all data related to programme and project management.

- **MinIO:** An object storage system used for managing uploaded and downloaded files, such as project documents and invoices.

- **Elasticsearch:** A dedicated log storage and search system for storing audit logs and supporting future potential extensions such as analytics or monitoring features

All components are designed to run independently within separate containers, with the system distributed using a docker-compose file to facilitate easy installation, upgrades, and system maintenance. Programmes are encouraged to operate Jems in virtualised environments, making it easy to adjust resources based on system load.

5.1.2. Key Technical Features and Good Practices

Performance optimization was a fundamental principle during Jems' development. Key decisions included:

- Use of native SQL queries for complex data exports, significantly improving performance for large datasets.

- Parallel data streams and versioning tables to manage large volumes of applications and reports efficiently.

- A strong focus on backend performance in early phases to ensure future scalability, although at the expense of less initial investment into front-end flexibility.

Plugin System

Building on the positive experience with eMS, Jems allows for the integration of plugins to extend system functionality:

- Plugins can perform business logic operations during workflows (e.g., application checks, reporting validations).

- Plugins offer flexibility for programme-specific customisation without altering the core system, supporting both internal development and external third-party development.

Data Providers

Jems introduced data providers to enable external developments, allowing programmes and IT providers to extract data in a structured way. Although this solution brought more flexibility, it also introduced complexity in performance management and integration.

Interoperability

Jems provides a RESTful API to enable integration with external systems, addressing a key shortcoming of eMS. This allows programmes to automate tasks, integrate Jems into broader IT landscapes, and future-proof their system setup.

System Updates and Maintenance

The docker-compose deployment setup ensures that updates and security patches can be deployed consistently across different installations. Programmes can choose to manage Jems maintenance in-house or through contracted IT service providers.

Feedback:

Stakeholders widely praised the system performance and stability of Jems, especially in comparison to eMS, where peaks of user activity had previously caused operational disruptions. Jems' architecture and backend performance optimisations, such as the use of native SQL queries and parallel data streams, were seen as major improvements that contributed to the system’s robustness.

The introduction of a RESTful API was positively received and considered a significant enhancement for interoperability with external systems. However, some feedback suggested that the API could be further simplified and better documented, to facilitate easier external developments by programmes and IT providers.

The plugin system in Jems offered valuable flexibility for programme-specific customisations without altering the core codebase. Nonetheless, the technical decision to bundle all plugins into a single standard package made it difficult for programmes to selectively share or adopt individual plugins without further adaptation. A more modular plugin structure could improve maintainability and reuse in future iterations.

Finally, while Jems' use of MariaDB's system-versioned tables provided important functionality for application and reporting versioning, it introduced increased complexity for database maintenance, particularly in relation to daily system backups. This was identified as an area where improved operational guidance and documentation could further support programmes in managing their installations**.**

5.2. Quality assurance

Quality assurance was a critical focus throughout the development of Jems, shaped by lessons learned from the eMS system. It was widely recognised that the complexity of a joint electronic monitoring system - with its high number of configurable options and interdependencies - made it practically impossible to test every possible scenario manually. Therefore, Jems adopted a multi-layered quality assurance strategy that combined automated testing, manual testing, user testing, and continuous feedback loops to systematically improve system stability and reliability.

Jems embedded quality assurance deeply into the development workflow from the outset. Key elements of the Jems testing strategy included:

- **Quick Tests and Kick-Offs**: Before feature implementation, quick technical feasibility checks and preparation sessions between Business Analysts and developers were conducted on each user story to minimise misunderstanding and rework.

- **Code Review Process**: Every feature required two independent approvals before integration, following the "two-thumbs-up" policy to maintain high coding standards.

- **Manual Testing of features:** Each feature was manually tested by developers and BAs to ensure accuracy and usability.

- **Regression Testing**: Regression tests, both automated and manual, ensured that new developments did not negatively impact existing functionalities.

- **Automated End-to-End (E2E) Testing**: Automation was prioritised in critical system areas, particularly in financial functionalities such as reporting, control, and payment modules, where data integrity and accuracy were paramount. End-to-end tests were integrated into the continuous deployment pipeline, allowing new releases to be checked for regressions efficiently.

- **Pre-Release Testing by the project team**: Before each release, a dedicated manual test phase was conducted by Interact's internal team, ensuring that business requirements and user expectations were consistently met. This included functional testing, regression testing, and testing against defined acceptance criteria.

- **Test Days with Core Group**: In addition to internal testing, specific Jems Core Group Test Days were organised before releases. Programmes participating in the Core Group were invited to test the new functionalities in dedicated test environments. These sessions provided valuable feedback from real users in real-world programme contexts and helped identify edge cases that internal teams might not encounter.

- **Hotfix Management**: When critical issues arose post-release, a structured hotfix process was in place, with root-cause analysis performed during hotfix reflection meetings to prevent repeated errors.

Overall, the Jems quality assurance strategy reflected a balance between automation, manual inspection, and community-based user testing, ensuring a high-quality system despite the inherent complexity of a shared monitoring environment.

Feedback:

Stakeholders who had experience with both eMS and Jems consistently noted that Jems demonstrated significantly higher system quality and stability. While absolute bug-free operation remains impossible in such complex systems, the proactive investment in automated testing and structured manual validation contributed substantially to user satisfaction.

Particularly, the decision to prioritise automation resources for financial modules — where data accuracy is critical — was praised as a strategic move. However, it was also noted that reaching the optimal level of quality assurance required more development resources and effort than initially anticipated.

Programmes that chose to independently modify the Jems core code were reminded that deviations from the official codebase might impact system stability and void quality guarantees provided through the official release process.

5.3. User support

Supporting users throughout the lifecycle of Jems was a key element of the overall system strategy. Recognising the diversity of programme needs and technical expertise levels, multiple channels were established to assist users in installing, using, and adapting the system effectively:

- **Technical Support:** Each programme that signed a Jems license agreement was entitled to eight hours of technical support from Interact. This support was typically used for initial installation, updates, or troubleshooting issues related to system configuration.

- **Jems Manual:** A continuously updated, versioned, and web-based manual was made available via the Jems Portal. The manual was designed to be searchable, structured according to major workflows (e.g., application management, reporting), and served as the primary source of reference for system usage.

- **Tutorials:** To further ease the learning curve, video tutorials were developed alongside the manual. Tutorials were targeted at explaining system functionalities that were either particularly complex or not easily covered through text-based documentation.

- **Helpdesk Platform:** A dedicated Helpdesk system, fully integrated with Jira Service Desk, was set up for users to report bugs, submit ideas for improvements, and ask questions. Helpdesk tickets allowed structured tracking of issues and ensured transparency in the support process.

Feedback:

Users appreciated the availability of diverse support options and highlighted the Jems Manual as a particularly valuable resource. The structured, versioned, and searchable format of the manual, updated after each release, was seen as a major improvement over previous approaches.

The Helpdesk platform, integrated with Jira Service Desk, was positively received for its professionalism and responsiveness. However, several areas for improvement were identified:

Users would have preferred the option to report bugs directly from within the Jems application, avoiding the need to switch to an external platform.

While bugs reported through the Helpdesk were handled diligently, there was a lack of transparency regarding when fixes would be integrated into upcoming releases. Even approximate delivery timelines would have been appreciated.

Some stakeholders suggested that greater visibility of reported bugs and submitted ideas could improve efficiency, by allowing users to see if an issue had already been reported or addressed.

Feedback and ideas collected through Helpdesk and Test Days was valued, but clearer communication on how user feedback influenced the roadmap would have strengthened the participatory approach.

Regarding tutorials, experience showed that creating video content was resource-intensive relative to its impact. Tutorials were particularly helpful for newcomers and for complex workflows not easily described in the manual. It was found to be good practice to produce video tutorials selectively for features where additional explanation was most needed.

Overall, the combination of a live manual, targeted tutorials, and structured Helpdesk support contributed positively to user satisfaction. Further integrating feedback mechanisms and enhancing communication about ticket and roadmap status would offer additional improvements in future development cycles.

5.4. Jems License

The usage of Jems is governed by a dedicated License Agreement signed between each participating programme and the City of Vienna, the hosting institution for Interact office Vienna. The License Agreement defines the legal framework for accessing, using, and modifying the Jems software, aiming to balance open access with the need to manage legal risks and maintain community coherence.

Key Elements of the License Agreement are:

- **Free of Charge:** Jems is made available to Interreg programmes without license fees, supporting broad uptake and shared development across the community.

- **Liability Limitation:** The License Agreement reduces the risk exposure for the City of Vienna by clearly stating that any malfunction or incorrect operation of the software remains the responsibility of the individual programme operating it.

- **Obligation to Communicate Code Changes:** Programmes that make modifications to the Jems source code are required to inform Interact of these changes. This ensures that significant adaptations are known to the community and can be considered for broader integration if beneficial.

- **Possibility to Request Contribution:** Interact retains the right to request that programmes, making useful changes, contribute their modifications back to the common codebase for the benefit of all users.

- **Jems Support:** Jems support is provided on a best-effort basis by Interact. No formal service-level agreements (SLAs) are established, reflecting the cooperative nature of the initiative.

Overall, the licensing approach adopted for Jems has been a key factor in enabling widespread adoption of Jems while maintaining sufficient control over quality and risk. In total, 48 Interreg Programmes choose to adopt Jems as their monitoring system, allowing them to avoid the costs associated with developing individual software solutions and collectively saving more than 15.3 million Euro in development costs across the Interreg community.

Feedback:

Stakeholders appreciated that Jems was made available free of charge and valued the overall cooperative spirit of the licensing model. However, some feedback indicated a desire for an even more open structure:

Several programmes and external IT providers expressed interest in adopting a general open-source license model (such as MIT or GPL licenses), which would formally position Jems as a fully open-source project.

Additionally, there were suggestions to establish an open-source repository (e.g., on platforms like GitHub or GitLab) where all programmes, developers, and external IT providers could directly contribute to the Jems source code. Stakeholders felt that this could foster greater community involvement, streamline the integration of additional developments, and enhance transparency.

At the same time, concerns were noted regarding quality assurance and governance: any transition towards an open repository model would need to be carefully managed to avoid fragmentation of the codebase, ensure rigorous quality standards, and maintain legal clarity regarding responsibilities for code changes.

Future considerations could explore possibilities for greater openness, provided that clear structures for governance and community contribution are put in place.

5.5. Meeting the requirements

The requirements for Jems were drawn from several sources, ensuring that the system complied with legal obligations while responding to practical programme needs. Requirements came in the following order of priority:

- **EU Regulations and other legal documents:** Core requirements were based on the Common Provisions Regulation, Interreg Regulation, and other relevant EU legislative and guidance documents governing the 2021–2027 programming period.

- **HIT (Harmonised Implementation Tools):** Additional standardisation requirements, such as templates for application forms, reporting structures, and control documents, were incorporated based on the HIT package to facilitate harmonisation across Interreg programmes.

- **Jems Core Group:** Programmes participating in the Core Group were instrumental in refining and prioritising functionalities beyond strict legal and HIT requirements, based on operational needs and user experience.

- **Jems User Group:** Input from the broader User Group, particularly collected through sprint reviews, Helpdesk feedback, and Test Days, further shaped the refinement of features, usability improvements, and additional developments.

Throughout its development, Jems remained strongly aligned with these layered requirements.

The robustness of Jems' compliance with EU legal requirements was independently confirmed through an **external audit performed by PwC in 2024.** The audit applied the international assurance standard ISAE 3000 and assessed compliance with key regulations, including Regulation (EU) 2021/1060 (Common Provisions Regulation) and Regulation (EU) 2021/1059 (Interreg Regulation). The audit tested critical control areas such as data storage and security, user authentication, audit trails, monitoring, financial management, and reporting functionalities. PwC concluded that Jems had been properly prepared in accordance with the relevant EU regulations, with no exceptions noted in the tested areas. This independent assurance confirmed that Jems provides a reliable electronic monitoring system fully aligned with legal obligations, offering a strong foundation for programme management and audit compliance across Interreg programmes.

To assess user satisfaction with Jems, several surveys were conducted among Jems license holders (in 2023 and 2024) and end users (in 2024). The survey results indicated high levels of satisfaction, particularly regarding the system's reliability and usability. The Jems User Feedback survey confirmed that over 80% of beneficiaries considered Jems to be easy to use.

Feedback:

Stakeholders widely recognised that Jems successfully covered the regulatory and HIT requirements necessary for the functioning of Interreg Programmes. However, several important observations and suggestions for improvement were noted:

Many programmes requested additional configurability beyond the strict legal requirements. There was a strong desire for greater flexibility, for example to adapt HIT structures where necessary to better fit specific operational realities. This often meant asking for customisable workflows or additional data fields, allowing for tailored adjustments while still supporting harmonisation principles.

Feedback from both the Core Group and User Surveys indicated that, while HIT templates were a valuable starting point, they were sometimes perceived as too static or rigid in practical implementation.

Programmes occasionally found that the standardised HIT structures did not always logically align with the realities of project management or control tasks within their specific contexts.

Some stakeholders suggested that a greater separation between mandatory fields and optional fields in Jems would have been beneficial, enabling programmes to configure the system more flexibly without creating compliance risks or changing core code.

Overall, Jems was successful in delivering a system that met regulatory, harmonisation, and most programme operational requirements. The experience highlighted the ongoing tension between the goals of standardisation and the need for flexible operational adaptation. Future developments could benefit from even greater configurability in non-regulatory aspects to better serve the diverse needs of programmes while maintaining overall coherence.

6. Conclusions

The development and implementation of Jems mark a significant milestone in the digitalisation of Interreg programme management. Built on the lessons learned from the eMS system, Jems successfully addressed previous shortcomings by delivering a robust, future-proof, and user-centred monitoring solution for the 2021–2027 programming period. Through a strong commitment to agile methodologies, open stakeholder involvement, structured governance, and high technical standards, Jems managed to meet both regulatory and operational requirements while providing a scalable platform for the future.

Key achievements include:

- a high adoption amongst programmes with **over 50% of Interreg programmes using Jems**,

- **15.3 million Euro saved by Interreg in software development** if those programmes would develop their own system,

- major improvements in system stability, performance, and interoperability, validated by high satisfaction rates recorded in user surveys with **over 80% of beneficiaries finding Jems easy to use**.

The collaborative development process, with active involvement of the Core Group, User Group, and broader stakeholder community, proved instrumental in ensuring that Jems evolved in line with the real needs of programmes. The combination of agile responsiveness and community-driven decision-making - particularly during the additional development phase with the introduction of the wishlist voting - fostered strong ownership among users and contributed to the system's broad acceptance. Jems prioritised quality through a combination of automated, manual, and programme-led testing.

At the same time, the project highlighted the ongoing tension between the goals of harmonisation and the desire for operational flexibility. While the reliance on standardised templates such as HIT was crucial to making Jems feasible as a shared system, feedback indicated that future developments should seek even greater configurability in non-regulatory areas to better accommodate programme-specific needs.

Looking ahead, the sustainability of Jems beyond 2029 will require continued governance, proactive investment in maintenance and adaptation, and potentially new models of community contribution, including consideration of more open-source approaches. Maintaining transparent communication with users will remain key success factors.

Jems stands as a successful example of collaborative software development in the European funding context. Its creation has set a benchmark for future projects aiming to balance efficiency, harmonisation, flexibility, and user empowerment within complex multi-stakeholder environments.

Annex 1: List of abbreviations

ERDF	European Regional Development Fund
EU	European Union
EC	European Commission
MC	Monitoring Committee
MA	Managing Authority
JS	Joint Secretariat
HIT	Harmonised Implementation Tools
Jems	Joint Electronic Monitoring System
eMS	Electronic Monitoring System
PO	Product Owner
BA	Business Analyst
UX	User Experience
QA	Quality Assurance
E2E tests	End-to-End tests