DOI: 10.2352/issn.2168-3204.2016.1.0.131

Developing Goobi - An Open-Source Workflow Tracking Tool for Digitization Projects

Steffen Hankiewicz; intranda GmbH; Göttingen, Germany

Abstract

The workflow tracking tool 'Goobi' is an open-source application designed to facilitate the management of small and large digitization projects in libraries, archives, museums, and other cultural establishments. It has been under continuous development since 2004. The flexibility with which Goobi can be integrated into everyday work routines and its potential to automate many digitization processes have made it an extremely popular solution.

Nowadays Goobi is used by a large number of institutions in many countries to produce millions of digitized items – together with standardized metadata. As software developers, we therefore constantly face new challenges. These include continuously adapting and extending Goobi in response to cultural and linguistic differences, the enormous range of project goals and data volumes, the sheer diversity of institutions and materials, and the minor problems that inevitably arise in all digitization projects. At the end of the day, there is of course so much more to digitization than creating sharp photographs.

Understanding the challenges of digitization

In the German city of Göttingen, we began developing Goobi, an open-source tool that allows users to coordinate their digitization projects, as long ago as 2004. [1] Although the original idea was to produce an application solely for a project set up by Göttingen University Library's Digitization Center, it soon became clear that Goobi could be used to avoid many of the problems that typically beset other digitization projects by clearly organizing the workflow. The key appears to lie in having a clear understanding of both the objectives and the challenges of the project. Numerous conversations with project managers at digitization centers in various countries show that the same challenges repeatedly emerge in their daily routine:

- coordinating multiple concurrent projects
- working with a range of workflows designed to produce different results
- relatively large numbers of project staff located in different rooms, departments, buildings, and even sites
- a huge variety of sources from which existing data need to be obtained (metadata, digitized material)
- different objectives, requiring that results be delivered in a variety of formats at the end of the workflow
- different types of material with specific requirements and individual metadata and structure data that need to be indexed
- a large and diverse pool of scanning hardware from different manufacturers
- unnecessary manual correction loops because errors are identified too late, thus creating additional work.

In almost every single digitization center, this list of typical challenges is matched by a wish list of efficient methods and ideal results:

- wish for greater efficiency when performing individual workflow tasks
- maximum automation of the workflow
- standardized results
- clear overview of progress for each object
- facility to check each team member's responsibilities
- effective quality control of results
- secure, centralized storage of data with simple arrangements for data maintenance
- statistics and reporting for projects, deadlines, costs, and billing.

As well as these requirements, users often say they are looking ideally for an inexpensive yet highly individual and user-friendly solution that draws on or integrates existing third-party applications for reasons of synergy. Naturally, they also want their chosen solution to interact with a variety of long-term storage systems so that they can be confident that the data they have laboriously compiled over the course of various digitization projects has been securely and reliably archived.

It was this wide-ranging set of requirements, and the need – articulated by many users – for a solution that can be easily integrated into existing architectures that led us to develop an open-source program that would offer the greatest possible flexibility and scope for expansion. [2]

The organizing power of workflows

In order to meet these requirements and provide the range of functionality described in the above wish list, we decided first of all to record and analyze the methods used in earlier book digitization projects.

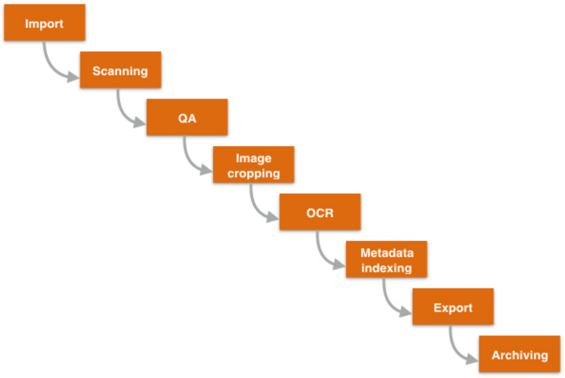
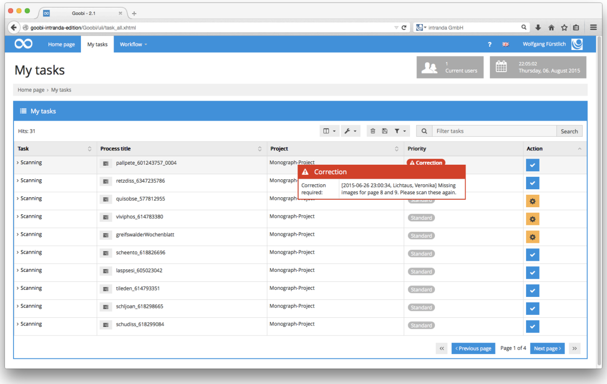


Figure 1. A workflow consists of any number of tasks that can be performed manually or automatically at a specified point.

DOI: 10.2352/issn.2168-3204.2016.1.0.131

We soon identified a series of relatively short but frequently occurring steps. These provided the crucial momentum for us to design a workflow management system that offered maximum flexibility. In order to process digitization workflows efficiently, the objective should be to break down all the tasks involved into units that are small enough to be coordinated and completed as sequential tasks.

Having established this principle, we were able to strictly divide up the various tasks, responsibilities, projects, and roles in Goobi. While some project staff initially felt restricted by the definition of a tighter structure, this new approach, largely based on responsibilities, sequences, and validations, quickly emerged as a guarantor of quality and therefore as a huge boost in terms of efficiency. Task lists, which are specific to each member of the project team based on roles and authorization levels and can be worked through in the form of a daily ‘to do’ list, helped to draw attention away from the more complex aspects of a project, and thus prevent common but unnecessary errors.



My tasks

My tasks

Hits: 31

Task	Process title	Project	Priority	Action
› Scanning	pallpete_601243757_0004	Monograph-Project	Correction	✓
› Scanning	retzdiss_6347235786	Monograph		✓
› Scanning	quisobse_577812955	Monograph-	Standard	⚙
› Scanning	viviphos_614783380	Monograph-Project	Standard	⚙
› Scanning	greifswalderWochenblatt	Monograph-Project	Standard	⚙
› Scanning	scheento_618826696	Monograph-Project	Standard	✓
› Scanning	laspsesi_605023042	Monograph-Project	Standard	✓
› Scanning	tileden_614793351	Monograph-Project	Standard	✓
› Scanning	schljoan_618298665	Monograph-Project	Standard	✓
› Scanning	schudiss_618299084	Monograph-Project	Standard	✓

Correction

Correction required: [2015-06-26 23:00:34, Lichtaus, Veronika] Missing images for page 8 and 9. Please scan these again.

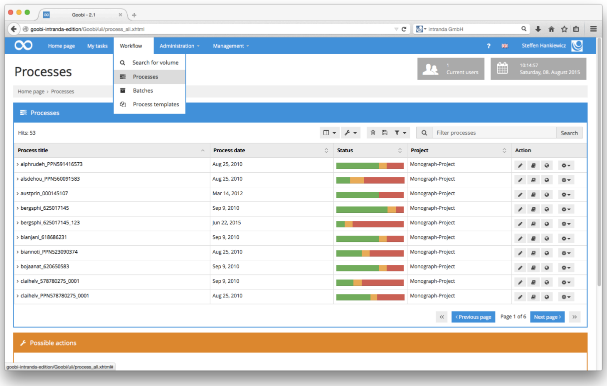
Figure 2. Individual workflow steps are assigned to project staff in the form of a ‘to do’ list that takes account of each person’s authorization level, role, and project membership.

Users were no longer distracted by unnecessary information, specific technical processes, or decisions about where to store the objects they had worked on. All these aspects of the project remained hidden behind the web-based GUI. Project staff could simply tick off their jobs one by one, usually in a strict order, and would only ever receive more detailed information if really needed in order to complete a certain task, or if the user’s role actually involved indexing that information in the form of metadata.

Standardizing workflows

Having adopted this new approach of splitting the workflow into small tasks, we quickly decided to introduce the concept of workflow templates. The idea was to generate a template at the beginning of a project, and then copy it as many times as necessary for every object being digitized in the same project.

The idea of specifying workflows using a range of freely definable templates for hundreds of objects, which – once generated in Goobi – would work their way through the workflow in a set order, ensured that each task could be completed in full and documented by the designated project worker at exactly the right point. This led to greater standardization and more consistent results. At the same time, the strictly defined methodology made it possible to store enormous volumes of project data using a centralized, standardized, reliable, clear, and low-maintenance structure.



Processes

Processes

Hits: 53

Process title	Process date	Status	Project	Action
› alphrudeh_PPN591416573	Aug 25, 2010		Monograph-Project	
› alsdehou_PPN560091583	Aug 25, 2010		Monograph-Project	
› austprin_000145107	Mar 14, 2012		Monograph-Project	
› bergsphi_625017145	Sep 9, 2010		Monograph-Project	
› bergsphi_625017145_123	Jun 22, 2015		Monograph-Project	
› bianjani_618686231	Sep 9, 2010		Monograph-Project	
› biannoti_PPN523090374	Aug 25, 2010		Monograph-Project	
› bojaanat_620650583	Sep 9, 2010		Monograph-Project	
› claihelv_578780275_0001	Sep 9, 2010		Monograph-Project	
› claihelv_PPN578780275_0001	Aug 25, 2010		Monograph-Project	

Figure 3. Goobi management overview to monitor the progress of workflow steps for all individual items in the digitization project

By following this approach, Goobi was able very simply to bridge the gap between the frequently voiced requirements of many digitization projects and their corresponding wish lists:

- simple and efficient working methods for users
- a clear overview and guaranteed results for project managers
- secure, centralized, backup-friendly, and low-maintenance storage for administrators.

Despite some initial hesitation, many cultural institutions in Europe came to recognize the potential of this approach and chose to join the Goobi user community, especially when we began to integrate a series of automated, CPU-intensive functions (e.g. for scripts, image processing, OCR, and rule-based indexing of standardized metadata in METS/MODS format).

Ongoing development driven by expansion

After more than twelve years of ongoing development, and adoption by over fifty institutions in nine countries (Germany, the UK, Israel, Austria, Switzerland, Spain, Denmark, the Netherlands, and Liechtenstein), Goobi is now almost unrecognizable compared with the earliest versions. Although Goobi was designed right from the start as a multilingual, web-based application, it became clear that a monolithic infrastructure would soon reach its limits faced with the diverse everyday needs of users working for different institutions with varying project aims, working methods, languages and cultures. In this context, it is interesting to note in our globalized world the influence of factors other than differences in the technical requirements of individual establishments (e.g. the variety of catalog systems that need to be integrated into the program). As developers of freely available, open-source software, we have also had to deal with legal obstacles and new challenges caused by differences in national legislation, for example with regard to the protection of personal data.

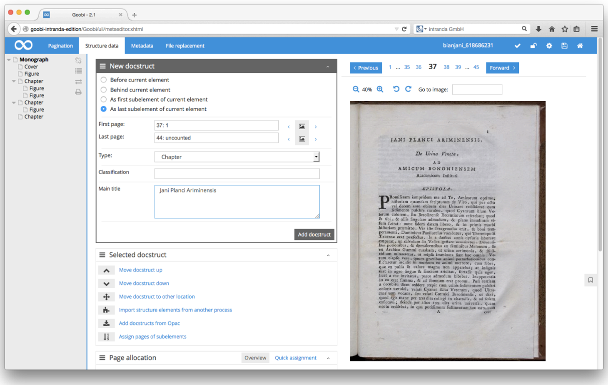
In response, we decided early on to revise Goobi’s architecture in order to produce an application with a more compact core, but with a host of plugin options to extend its functionality and thus cater for every possible approach to digitization. Most of these plugins are free and can be installed

DOI: 10.2352/issn.2168-3204.2016.1.0.131

individually wherever additional functionality is wanted or required. There are plugins for just about every conceivable purpose – from data import and export through to special editors, automatisms, and statistical reports. [3]

Users and the Goobi community

Thanks to these plugin extensions, Goobi has since been used with great success to coordinate every conceivable type of project – from modest, two-person operations with fifty digitized books through to major projects with a hundred staff working simultaneously and mass processing across several distributed systems. You can find almost identical workflows in many digitization centers (e.g. involving retro-digitization and export to a presentation system). By contrast, other workflows are so specialized and designed for a particular infrastructure that it would be very difficult to adapt them for use by another institution (e.g. on-demand digitization with a dedicated and integrated payment system).



New docstruct

○ Before current element
○ Behind current element
○ As first subelement of current element
● As last subelement of current element

First page: 37: 1
Last page: 44: uncounted
Type: Chapter
Classification
Main title: Jani Planci Ariminensis

Add docstruct

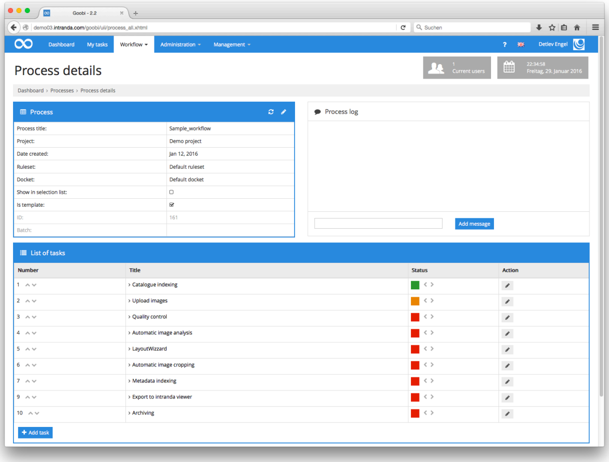
Selected docstruct

- Move docstruct up
- Move docstruct down
- Move docstruct to other location
- Import structure elements from another process
- Add docstructs from Opac
- Assign pages of subelements

Page allocation — Overview — Quick assignment

Figure 4. Goobi's embedded METS Editor allows users to create logical structures and enrich metadata in standard formats.

Given the sheer variety and ever-growing number of plugin interfaces available to all users, there are practically no limits on functionality as far as the technical infrastructure is concerned. In terms of functionality, this means that the direction taken by Goobi within an establishment can be determined primarily by users themselves, who are always keen to swap ideas on future projects, methods and features at national and international gatherings. Discussions at such meetings cover subjects as diverse as the modifications required by specific establishments or how major developments can be planned and funded. The crucial point is that both small and large institutions come together as equals to exchange ideas on methodology, the problems they encounter, and the new features they would like to be included, without the future development of Goobi being dominated by large institutions alone. Thanks to this cooperative approach, and because the results of ongoing development work are generally available to all users without paying a license fee, Goobi has been adopted by numerous small and large establishments in the library sector.



Process details

Process

Process title:	Sample_workflow
Project:	Demo project
Date created:	Jan 12, 2016
Ruleset:	Default ruleset
Docket:	Default docket
Show in selection list:	☐
Is template:	☑
ID:	161
Batch:	

List of tasks

Number	Title	Status	Action
1	› Catalogue indexing	🟩	✎
2	› Upload images	🟧	✎
3	› Quality control	🟥	✎
4	› Automatic image analysis	🟥	✎
5	› LayoutWizzard	🟥	✎
6	› Automatic image cropping	🟥	✎
7	› Metadata indexing	🟥	✎
9	› Export to intranda viewer	🟥	✎
10	› Archiving	🟥	✎

Figure 5. Workflow templates can be individually configured, contain both manual and automated tasks, and be applied to any number of objects in the project workflow.

It is worth noting that a coordinated workflow-based approach can prove valuable much more quickly than generally anticipated. At what point then is it worthwhile for smaller projects to employ a workflow tracking system? The key factor here does not appear to be the number of objects being digitized. In fact, the need for effective coordination becomes clear once several tasks need to be performed for each object. Above all, however, this need arises as soon as the project involves several users or user roles, since this creates dependencies between those persons and between their respective tasks, which now have to be performed in a certain order for each object. As a rule of thumb, if a project involves more than fifty books or more than three colleagues, it is worth using Goobi for coordination and quality control purposes, and to maintain an overview.

Lessons learned

Developing a professional, multilingual open-source application for the international cultural heritage sector is much more demanding than you might initially expect. Excitement about the successful expansion of the user community is tempered by the growing requirements of a complex user landscape. Technical demands and the task of mapping and indexing bibliographical metadata pose significant challenges, as do linguistic differences and cultural variations in the working practices of Goobi users. On balance, these tend to lead to a more robust program. The following list summarizes the main lessons we have learned in recent years as Goobi developers:

- With regard to digitization projects, the challenges facing small institutions are no less important. In fact, they are often able to cooperate more responsively and show greater flexibility when looking for a solution.
- In our experience, it is not possible to provide and regularly update comprehensive multilingual documentation for users in the form of a wiki. Even in 2016, users expect handbooks that they can print out.
- Every newly implemented function should be accompanied right from the beginning by a configuration option that allows

DOI: 10.2352/issn.2168-3204.2016.1.0.131

users to deactivate it. Even better, develop an extendable and robust plugin infrastructure at an early stage. This is the only way you can integrate the new features requested by users without jeopardizing the stability of the software core.

- It is a good idea to maintain strong links with at least two establishments so that you can arrange for any new version to be tested in productive form for several weeks before it is finally released. Even after these tests, never install a new version at the same time on too many user systems, even if it appears to be stable.
- Never modify or remove an existing function. There will always be at least one user who wants to keep it in its existing form.
- If possible, use your own software under productive conditions. That way you can learn a lot about the software and discuss how to use it from a much more genuine perspective. As the developer, you also benefit by spotting any faults much sooner.
- It is vital to arrange a meeting of users at least once a year. Allow roughly 50% of the time for presentations on methods and new features and the remaining time for users to share their knowledge and experience between each other. This is the only way for developers to find out how their software is really being used and what users want.

Outlook

Just how reliable Goobi's infrastructure is in productive operation has been amply demonstrated over the last twelve years, during which it has been used by numerous institutions to generate millions of digitized items. As well as guaranteed trouble-free working, however, Goobi offers additional functionality in the form of existing plugins and the potential for innovation through the development of new ones. Thanks to a fast-growing user base that produces a constant stream of fresh projects, we have the ideal breeding ground for new ideas. While high-quality OCR and crowd-sourced proof-reading of text in the resulting images are now commonplace in the field of digitization, for example, we are still in the early development and testing stages when it comes to efficient methods of transcribing handwritten sources, smart image analysis, automatic metadata enrichment for named entities with persistent links in standardized data formats, and the semantic analysis of full text output.

As a workflow management tool, Goobi's basic job is to coordinate the production of digitized material and provide an overview of the progress made during each project. As we continue to develop the software, however, our everyday focus is also on ways in which we can expand its functionality by integrating new and innovative tools and thus help other branches of research to work even more effectively with the digitized output and corresponding metadata. Both our own innovations and external solutions can and should be incorporated into digitization project workflows in the form of plugins. This is the only sustainable way in which the contents of millions of works can be digitized and made available for downstream analysis by researchers.

As Goobi developers we face new challenges every day in response, for example, to constantly changing standards at metadata level (e.g. RDA) or the currently observable trend towards cloud-based IT services and (of particular importance in the context of digitization) the associated demand for efficient transfers of data to remote servers within workflows. As soon as we begin planning a roadmap for the next version of Goobi, no doubt we will be reflecting on the increasing importance and requirements of 'digital humanities'. Above all, however, we will talk to our users.

References

[1] http://www.intranda.com/goobi

[2] https://github.com/intranda/goobi

[3] http://www.goobi-marketplace.com

Author biography

Steffen Hankiewicz is a senior software developer, CEO and owner of the German software company intranda GmbH. He has been developing and implementing software solutions for digitization projects for more than 15 years now. The open-source workflow management software Goobi, the proprietary intranda viewer as well as an automated TaskManager for OCR, JPEG2000 conversion or Named Entity Recognition jobs are some of the current digitization tools he develops and supports together with his team.