https://doi.org/10.2352/issn.2168-3204.2017.1.0.78

Automation in (mass) digitization QA-workflows

Martina Hoffmann; National Library; The Hague; The Netherlands

Abstract

In setting up a QA workflow – or any other type of workflow - one tries to make processing faster, better and more efficient. As we are often dealing with vulnerable originals, work on those documents can only be automated to a certain extent, but within the scope for automation, all opportunities should be used. Based on the example of the Netherlands large digitization program Metamorfoze (specifically the Archives and Collections section) this paper will give an example on how to achieve such optimum automation for QA control on data-integrity and will try to answer key questions on automation as they are the starting point for a better QA-workflow.

Quality Assurance challenges in Metamorfoze Archives and Collections

(Mass) digitization projects are done all over the world in different settings and with different requirements and output. In the Netherlands Metamorfoze is the national conservation program for cultural heritage on paper. Within Metamorfoze there are two distinct sections: one for books, magazines and newspapers and one for the remainder, called the 'archival section'. In a project of the archival section up to seven different parties may be involved, all with their own requirements and wishes. Some are more prominent than others and some are rather 'nice-to-have' than 'must-haves'. The more variables we meet the more complex a process gets usually and with several (external) parties involved communication can get fuzzy. Not to mention the unique often very fragile objects that we encounter in the archival section where no page seems to be the same within any given project. How can we deal with such a variety of needs in QA? After establishing the need for a profound quality control system in order to deal with the 'must-have' part of the projects we went on with setting up a workflow that is both flexible enough to meet the different needs and structured enough to provide a generic framework in order to be able to automate most of the QA steps involved. The need to automate several steps comes with the sheer volume of data we are working through every year: approximately 350 TB within the archival section QA-workflow. Putting all requirements together and simplifying it somewhat for clarity's sake our QA-workflow right now consists of four large process steps: data integrity, image quality, preservation masters and access copy (see fig. 1 below)

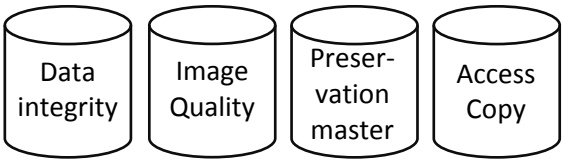

Figure 1. Four step process for Metamorfoze archival digitization projects

Due to the fact that we do only ensure the quality of the delivered products at the National Library our main focus is ensuring the quality of the first three steps which do include ensuring the data is suitable for long term preservation at the digital repository of the National Archive of the Netherlands. The fourth step 'access copy' is a check on content for instance if every page of a given unit is digitized or if the special instructions are followed accordingly. This step is carried out by the collection owners who have the contextual knowledge of the archive. They are also responsible for the digital images to be made available online or offline on site.

The history

In the early years we began with much smaller amounts of data per year and so initially, our QA-workflow contained a lot of manual steps and some weren't even included in the process due to specialists who were fixing eventual problems on different places in the process line. For example: we checked samples of files on required header information, putting filenames in databases or excel sheets, which were controlled by hand. There was no check on inventories as the inventories were created in a step after the quality control by another department with its own specialism. In the case of Metamorfoze archival and collections this was especially tricky because the collection owners provided different sets of inventories according to their own available software and sometimes even different versions of inventories within one project.

The downsides of this approach are: time consuming and the capacity of an employee required. Furthermore there are more departments and specialists involved which means more communication lines, more working pressure on each specialist and way more deadlines to keep in mind and organize. In quality control processes the time of employees can be used more effectively and doing everything by hand means of course the level of potential errors (things are overlooked, no 100 % checks) is high. Automation can answer to a lot of needs in such a process. From the basic scripts we already used like checksum validations or batch files to create lists we evolved our control to more scripts and thus more automation. We did make for each check a separate script which was quite logic in de process of learning what we could automate. It helped us to test software for specific parts of checks and making sure that the software we use is better than an employee doing the same thing. This step takes a lot of shopping around and talking to experts, software engineers and time to test.

While it is tempting to take a separate script for each separate check as we had, it has a serious downside: starting each script separately takes valuable time to start and set up according to parameters for each collection. Furthermore most of the scripts have to be run in a certain order. Some scripts would just not perform if any other script was run at the same time as well as some criteria were (and still are) what we call "knock out" criteria. Fail to meet the "knock out" criteria and we have to reject the

batch of data e.g. checksum errors or null byte files. Also we did suffer from insufficient processor capacity when running several scripts at the same time. Time lost in starting the separate scripts also had a knock-on effect on other checks: measurements of the targets and checks of samples of the actual images had to wait for completion of the scripts containing the data tests. In order to speed up this partial process we did a lot of benchmarking in different programming languages and with several tools that execute the same controls. In all that we have mainly used *.txt files for a long time as output results which we then (re)viewed in programs like UltraEdit or Notepad+, other text viewers and of course the excel sheet import - first with row limitations in older versions but later on we could actually import a complete batch of header information without having to split the *.txt in several parts before importing it. That already was a huge improvement but nonetheless importing data in excel sheets does have its problems with formatting as is widely known. Especially values such as ISO or shutter speed which are commonly found in file headers of digitized material gave faulty results after import. But it was cheap, available and workable. Nevertheless we realized that this was only working as a temporary solution as the output as txt proved to be a handy tool, but once we started controlling hundreds of gigabytes of data they became outdated nuisance. In the meantime we kept searching for better, faster more usable tools and easier reportings.

Most of the scripts we tested at that time were scripts written by (internal) software engineers in java, python or perl. As we did handle and had to configure several different scripts manually before starting we first wanted to see if we could minimize the effort to start a data integrity check for the employees by combining all the different scripts to be configured and started in one step. With the help of a software engineer, we created a *.bat file with a corresponding config.ini file in which we could set all the parameters for a certain unit of data. In order to do this we had to prioritize the checks in sequence and also state what the parameters for each check were as well as which parameters could differ in each project. With that step we cut the time an employee needed to manually put all scripts to work from half an hour per batch of data to approximately five minutes. It was a huge success for freeing up QA-capacity. However, what we did not accomplish was minimizing the time a computer needed to perform all those tasks and reducing the time it took an employee to read the output. The journey had only just begun.

Automation of data integrity checks

The main challenge we faced were the big challenges in all QA-workflows especially in digitization being: Time - Money - People. At any given time we have to ask ourselves the three key questions:

- Why do we need to check quality?
- Why does it take so long?
- Do you know how much it costs (time & money & people)?

In order to answer to the managements needs improvement is necessary and of course we want it to be fast, efficient and low cost in the end. Easy to maintain, fast to implement and efficient in performance and results. In our case: With the first question answered by the outlines of the national program - mass digitization as way for mass conservation according to the Metamorfoze Preservation Imaging Guidelines [1] one could argue that QA on image quality was enough. Actually nothing is less true in digitization. While image quality does need to be checked the images themselves need to be checked for long term preservation usage and without metadata even on the image data we will not be able to use the generated digitized content for long. This is especially true for the archival section where there is no widely used standard for metadata. Descriptions can vary per project or even for a collection within a project. They vary by institution and as our partners are not all archives but also museums and libraries, they all use their own system to describe their collections. So in order to ensure that the National Archive can manage those data in their repository we have to find or create a common denominator for all those variables.

Two distinct factors played in our favor at that time: First - The National Archive of the Netherlands works with a XML-scheme for ingest of their own data into their repository. As we have to put the data from the archival section of Metamorfoze also in the same repository the data we provide has to follow the same XML-scheme. Second - The department of specialists that used to do all this work for us was no longer available to us and we had to find a way to do the same work without specialists. Therefore we created an automatic way to ensure all the data is captured the right way in order for long term preservation:

Figure 2. Inventories check start screen

Inventories give us the least amount of data that is needed in order to manage the long term preservation besides the filenames themselves. They deals with ownership of the collections as well as access rights and publication limitations. We identified this as being the common denominator of all the collections we were dealing with and thus set up a pre-QA step. Pre-QA because it has to be carried out before digitization even starts by all collection owners who wish to digitize their objects within the Metamorfoze archival section. In essence, we translated the XML-scheme that the National Archive uses for ingest into a simple excel template with mandatory or optional information about the collection and each object. The collection owners who are then obliged to send those excel sheets to the National Library where we will run a script in order to validate the information technically for e.g. unique object numbers, no special signs or if access information is

provided. in the same step our script will convert the excel sheet into a XML-file which is basically the same XML-scheme that the National Archives will be using for their long term preservation ingest. This gives us a number of advantages:

- We are now able to make sure that all parties involved work from the same starting point
- We can cross check all incoming data to the provided information
- The XML-sheets don't have to be created by specialists at the end of a project
- The data can be validated before digitization in order to prevent non unique filenames or incomplete access rights.
- For the digitization party it can be used as checklist of inventory as well so their logistics have been adapted as well which resulted in less discussions about whether objects have been transported where and when.
- Putting all the necessary information in one file and send it beforehand to the digitization party enables them to make barcodes so we can reduce faulty filenames and make sure that the objects are linked to the correct inventory number.

Having a script up and running as easy as one can see in figure 2 with essentially three options has the advantage that we only have to add a location of an excel sheet to the script via 'add inventory (style x)' and the output XML is automatically stored in one location with time and date stamps which means that by one-click we also get version management on inventories and we do not need to administer them separately:

Figure 3. Example screen list of inventories

We have to be honest here: it took some time to explain to the collection owners what the benefits of such a process are. Fortunately we got feedback and did improve the sheets accordingly. We do provide a concise instruction manual for collection owners to explain how and why they need to fill in the information we need. So far we are getting great responses from the digitization parties and also the collection owners, who are now familiar with the process. The biggest benefit they get out of this step is that we can encounter double inventory numbers in their catalogs which they did not know exist. So they can also correct faulty data in their own systems.

By implementing this pre-QA step in our workflow we made it more mature also regarding the contributions of our external partners. Internally, however, it became evident that there was a need to do more checks on data integrity than the usual checksum, null byte files and filenames. Having set up the inventories pre-QA step, we now know what files to expect, so we wanted to add cross checks between the inventory and the actual files, as well as more extensive checks on filename and map structure. By logging all data we can prevent the occurrence of double filenames through the whole process and not just for one project at a time. To achieve this we needed more, better and faster software.

We still had our previous bunch of scripts that worked but they took far too much time to process and also a lot of time was needed to check the output. Looking for a solution, we did not come across any off-the-shelf software that implemented all our needs and wishes. Therefore the choice was made to integrate our existing scripts and make them faster. We accomplished this by translating all the scripts into one program language (perl) using existing extension packages for different checks. This had immediate benefit of speeding up the process, which recovered valuable processor time. In order to combine our new pre-QA step with the existing checks as well as expanding the range of those check we chose a strict modular approach in our scripting as well as in the user interface. As seen in figures 2. & 3. we did choose a simple GUI for our pre-QA step where a clear title tells the use of the page, a few simple options and a searchable overview in table form. What is not shown is that everything is installed and running via a local host on our system and can be easily accessed via a bookmark in the browser. It does work in all current browsers so every QA-employee can use his or her preferred browser. We did match this style to the rest of the data integrity checks we translated to perl which means that everything is now accessible from the same browser page and the real work is hidden for the user. The configurations have been built into the scripts and there are only little manual steps left for a QA-employee to start a batch control.

We clustered the modules to cater for the following checks and actions:

- checksums checks (1)
- checks on header information (2)
- check against inventory of expected files (3)
- check on filenames and folder structure (3)
- mapping files against targets (including time checks) (4)
- taking samples and storing the randomly chosen files in separate folders, for the QA-employee to start working immediately (5)

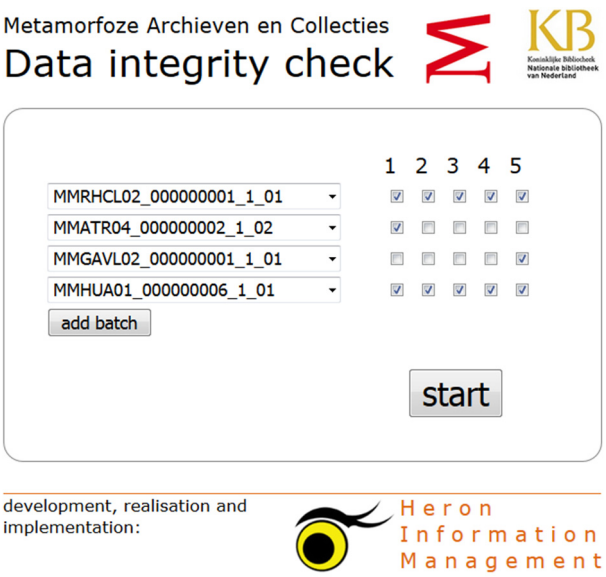

Figure 4. Start screen for data integrity checks

The five steps mentioned above are not single tasks, they are clusters of related tasks. For instance the checksum control includes checks existence of the file, the checksum and the null byte check. However putting them in five clusters allows for quick identification of the general area of any problems. Investigation in depth afterwards is not difficult. Also by putting them in a logical order we prioritized the breaking points in data integrity. For instance a faulty checksum or the discovery of faulty files will automatically stop the software generating a report with the errors and the batch will be rejected by our QA. The supplier will get the batch back and will be asked for a new delivery. As shown in figure 4 the screen itself is simple. One chooses a path with a batch and then clicks which modules do not have to run (by default all five modules will run) - push start and the process will take place. Results of the checks are recorded in XML and stored in an open source native XML database (eXistDB). While we create a lot of data in this process and even a lot of XML files we do not wish to read all the data in XML. For querying the results, a combination of perl and JavaScript is employed:

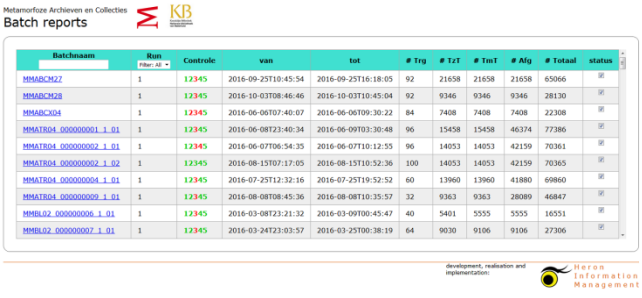

Figure 5. Start screen for data integrity checks

The most important thing for the GUI we wanted was flexibility. We started out with a GUI for our pre-QA step but we did not want several different stations for each new module of checks we implemented - so not only our back end approach (the actual scripts) had to be modular but also our front end (the GUI) had to be as modular. We therefore have a quite simple layout which can hold a dashboard to the two major steps and follow this layout throughout the whole data integrity process. This new approach with translating, regrouping of scripts has resulted in a total time consumption of 4 to 6 hours per 2TB (depending on the number of individual files), which is a reduction of at least 20 hours compared to what we needed previously using different software languages and manual configurations. Furthermore, we are now able to check the output of all the data per batch within 5 minutes instead of 4 hours even with more checks done. Putting a color system - red (error), green (valid), orange (attention) and black (fatal error) - to the result page makes it easy to read for our QA-staff and therefore fast. Furthermore we create now output reports (via query from the XML database) that only state errors with exact location and specifications of the error (see figure 6).

Figure 6. Example report created by KB (dutch version)

From this report we can browse further for more in depth overviews via links that automatically open in another tab of the browser or copy paste the results shown in the report formats we do send to our suppliers. The report we created in figure 6 shows for example that we could not match several images to their according daily target set which would result in a rejection of this batch. Besides this we have not the expected value for date/time in the header information in the expected XMP-tag (63117x DateTimeOriginal leeg). The empty XMP tag corresponding to 'copyright' is only a warning line (in orange) because we actually do expect that tag to be empty generally but sometimes suppliers will provide information as 'not available' in this field which is technically not incorrect.

Data that are faulty are not suitable for long term preservation, so they are not worth any further time of our QA-employees. The time intensive ‘looking at pictures’ (see fig. 1:

steps 2 and 3 in our QA-workflow: checking targets and scans) will only take place once the data checks have been passed successfully. The increased efficiency pleases us all, including our managers.

The future

Automating our QA-workflow in such a way that all data related tests take place at the start has already reaped huge benefits. However, we are not done yet! There are still lots of checks we want to implement, as well as steps to automate that we still do manually now because we did not quite translate the logic into code yet. For now we accomplished to compute and automate the most common variables although we encounter each project another variable we did not account for in advance by now. So the biggest challenge we are facing now is to translate more variables of already existing checks into the software to make it even more useful for all our projects and on the other hand adding news features and modules. The next steps will be, among other things, adding black pixel detection, further improved reporting and a module that can be used by suppliers for taking samples. However, we set ourselves as a goal that the total time to process a batch (2 TB of data) will never exceed 24 hours. We will continue to build modules that we can add to our base and that interact on a useful level with our report requests. Also we will try to identify more steps in our workflow that we can automate that way. There are limitations to this as the trained human eye still is the best in detecting artifacts and flaws in a digitized image.

Lessons Learned:

As illustrated above we have benefited a lot from reversing the steps of QA analysis and putting data integrity as our first step– in time and ultimately in money. Furthermore we benefited from circumstances that forced us to assess data even before digitization takes places. While automation takes time and costs money – these costs are far lower than the structural cost of an QA-employee, who is better used for things a computer can't take care of anyway. Of course we bought some computers with more processor capacity but in the end we gained many valuable hours of free capacity back in order to process more data each year. But there was much more to learn:

- The benefits also extend to our suppliers: once their data processing is set to meet our demands they can concentrate on making the best possible images. To accomplish that, we also provide them with background information regarding our requirements and ask them for feedback.
- The collection owner as well as the National Archive are provides with data of the digitized content. For a collection owner it can mean a cross check of their own catalogs and the National Archives gets a wrapped up package of data to manage their long term preservation duties of the digitized images.
- It is essential to provide the key tools like our inventory style sheets in order to keep the process simple for all external and internal parties.
- The key challenge is translating user requirements from different parties and seemingly endless variables into a few logical options and then translating them into code.
- We have also learned that automation (using scripts) does indeed save time, but that there are still many things to be thought through and put to the test before the optimum script can be written. The same goes for processor capacity and managing computer power. Benchmarking is essential.
- Breaking down the large process into small chunks and sort them by logical steps helps developing a modular database that can grow to meet new needs.
- A simple but flexible GUI greatly benefits the process and is easy to introduce, because everybody can read the information obtained unlike complex reports.
- We mine a lot more data about our productions than we use in our current process, but that means that we are prepared for possible future questions regarding our process.
- Communicate transparently with all parties and time put into explaining why you ask for the things you do is not wasted.
- It is impossible to think of every variable beforehand. Be flexible to change!

And last but not least:

- Once you start with automation you can't stop thinking of more things to automate in the process.

References

[1] Guideline Preservation Imaging Metamorfoze; https://www.metamorfoze.nl/sites/metamorfoze.nl/files/publicatie_documenten/Metamorfoze_Preservation_Imaging_Guidelines_1.0.pdf

Author Biography

Martina Hoffmann is Senior Production Manager digitization at the National Library in the Netherlands for the archival section of Metamorfoze. She was operational manager quality control of digitized products in the National Archives in the Netherlands. She co-designed several quality assurance workflows for different mass digitization projects in the Netherlands. Starting with only image quality QA processes her main focus now are QA processes including several fields of expertise from metadata to long term preservation.

