Form Management in Public Administration

Miguel Verdelho
miguel.verdelho@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

September 2020

Abstract

Some Portuguese Municipalities currently provide online most forms and documentation their citizens might need to perform any type of requests. Managing all the documentation and its provisioning is executed manually, without any type of supporting tool or validation. The way this management is performed is problematic when updating and distributing hundreds of forms. The system presented in this report aims to ease the delivery of sets of versioned documents through automation of tasks, a clear interface and error detection features. Its major benefits are the validation of human actions, automated control of versioned documents, data coherency when fetching up-to-date documentation and mostly the automation of document/packages delivery. This application provides some document management functionalities having the extra possibility of scheduling deliveries and gathering documents in sets to be automatically delivered. The goal was to develop a Web tool with new workflows that could ease the process of managing versioned forms as well as their availability for the population. In order to evaluate the gains obtained from the use of this application, tests were performed with a diversified universe of users who executed the same actions with and without it, giving a clear feedback on the improvements brought from its use.
Keywords: Process Dematerialisation, Document Management, PDF Forms, Web Application, Urban Management

1. Contextualisation

Process's dematerialisation revolutionised all types of businesses in the past few years. Digital documents are prevailing over physical files, not only because of environmental concerns but also for all the advantages they bring to organisations: cheaper storage costs and automation of document management processes.

Following a European Union's tendency, the Portuguese Government (2017) [1] issued a resolution in order to reduce public expense by cutting down paper usage costs and paper-related processes, investing in a modernisation of the public sector with a more sustainable and paper-free administration. Thus, software solutions for this type of administration with document management features keep increasing in popularity.

Portuguese municipalities are in charge of urban management which means that whenever a citizen needs licensing/information from the municipality, an application is made through the submission of a document. Usually, these requests are associated with different kinds of documents: forms, responsibility statements or certificates that may need to be submitted for a successful application.

The process of submitting applications through paper documentation used to take days or weeks, but the use of digital documentation allowed for a process dematerialization. Now-a-days, using PDF Forms (and other technologies), the process of fetching and submitting documents can be executed from any where any time.

ePaper is one of the applications used by Municipalities in Portugal on top of which this study is developed.

1.1. Problem Description

The current version of *ePaper* has a public platform for citizens to download and submit forms that allows limited configuration/maintenance from the municipality. The application is still not prepared to deal with this document versioning and the following deployment. Specific problems arise:

1. Providing a new online form demands a complex sequence of tasks, including pasting files in specific folders mapped by the application, with a exact name that *ePaper* needs to recognise.
2. Municipality workers need to perform complex replication and copy of files between the different systems.
3. Updating a form and making it available to the public is a liable procedure that includes replacing the version with a new file with the same name and generates many human errors. Besides, it keeps no history of form versioning.

Even though there is a way to go around the fact that the application does not support versioned documents (by executing manually part of the tasks needed), it is a very liable procedure that is associated with a large amount of human errors. There are strict rules in what concerns file management such as specific file paths and file names that users must follow on the replication process and there is not any kind of control or verification to assure the actions taken are valid.

1.2. Objectives

Given the previous contextualisation, this report aims to describe in full detail the study performed on this topic

as well as all the decisions taken that led to the solution found. The work presented in this report aims to achieve the following objectives:

1. A problem analysis giving an insight on why the current implementation is problematic.
2. A clear specification of what the project requirements are, within the scope of ***ePaper***, where it is meant to be integrated.
3. Construction of the web application that meets all the requirements.
4. Evaluation of the tool built and the gains obtained from its use.

Overall, the main goal of the work presented is to improve the management of versioned forms easing and simplifying the *ePaper* mechanism.

2. Background and Related Work

Within the context of this work, it is relevant to consider some concepts which are closely related with the topic at hands such as process dematerialisation and document management.

2.1. Process Dematerialisation in Public Services

Paper has been in use for information exchange for many centuries now. In the last two/three decades Information and Communication Technology allowed corporations and consumers to increasingly use technologies for many of the paper demanding tasks.

Public organisations are no exception in this evolution towards document digitalisation in the services they provide.The interaction between citizens and public administration began changing in some of its major aspects [2]. The means and settings in which citizens communicate with municipalities changed once it became an online process. The actors involved changed because requests can be automatically handled without a receiving actor. Besides, the initiation, duration and scope of these interactions also changed given the new paradigm of digital services [3].

Process dematerialisation, in this context, is the replacement of paper documentation for digital files as information carriers in order to eliminate the circulation of paper and its related tasks. The term *digital public services* is used to refer to the public services provided using internet based technologies where citizens may interact with organisations entirely through an application. One of its advantages is the implementation of this self-service concept where citizens should be able to demand data and services from anywhere.

Public administrations in many countries are taking this step of moving towards paper digitisation and process dematerialisation. Casalino et.al [4, 5], performed studies over the Italian public administration system and what should be the gains and losses of paper digitisation and process dematerialisation. Concerns and drawbacks of this transition are pointed, namely the constant re-investments in new technologies and in training people on how to use them, and all the actions needed to encourage the population for the use of newer systems. Despite this, conclusions are that those costs are small when compared to the advantages brought by innovative systems. They argue that benefits would be clear from an economical point of view (where new technologies lead to relevant savings) to an increase of HR productivity and a smaller environmental impact. With System Dynamics Models[6], their simulations show how, within a scope of just a few years, break even points can be achieved and administrations can profit from all the expected advantages.

For public administrations, technologies such as PDF may offer process dematerialisation in two major ways:

- **Information Distribution**: static PDFs generated by administrative software such as *ePaper* automate the diffusion of information as a consequence of a deadline expiration or any other notification. Citizens may fetch all kinds of information provided by the municipality in a form of a PDF and ease all the information gathering.
- **Automated Requesting and Processing**: digital Forms help eliminating paper usage by providing citizens with the possibility of remotely download, fill, sign and submit forms online. Not only this type of process becomes more comfortable to citizens as well as it may become much simplified for institutions that receive these requests: templatised documents can be read automatically and data can be processed, filtered and organised with less (or none) human intervention.

Besides these technologies, there are software solutions that provide process dematerialisation in this context of public administration. Diverse Electronic Document Management Systems are commonly used by public administrations in order to manage all the documentation they have to maintain, their versioning and lifetime among other things. Despite not being the focus of this thesis, Document Management systems, their properties and functionalities are studied ahead. On the other hand, there are Workflow management systems extremely useful inside organisations for the automation and process dematerialisation that they bring. All requests that any citizen submits in a digital format can be automatically processed by this type of applications with adequate workflows for each business bringing an end to all the paper circulation that was attached to this type of processing .

2.2. Electronic Document Management Systems

Electronic Document Management Systems are tools that organisations use to manage valuable information and documents. It can be looked at as a tool that eases and automates the processing that each document can be subject to as well as manages and relates all the data extracted from documents inside an organisation. Huge technological advances in EDMS led to a pivoting in every type of business towards paper-free business models for a diverse amount of reasons. These systems are gaining relevance in the market due to their several advantages [7] such as physical storage space reduction (through hard drives instead of warehouses), lower time consumption in documents

exchange and competitive edge over other businesses through an increase of efficiency in internal processes and workflows. For these and some other reasons, since EDMS were adopted by organisations, they became crucial tools for the maintenance and development of their businesses. Despite this, such evolution also presents some adversities[8, 9]. The high rate of technology changes and the desire and need of implementing those in organisations in order to make them more efficient and productive is a challenge that Information System(IS) managers have to face.

Bigger organisations frequently face the challenge of fragmented data where different departments handle data in different ways. EDMS will have the job of managing data across departments becoming a more useful knowledge source for managers and will eventually lead to performance improvements and easier organisational management. The challenge for Information Systems' executives is to integrate all the existing technologies and linking all the separated data into one unified system. This issue might grow in complexity as technology evolves unevenly in different but related areas [8] of action inside an organisation. If new and emerging technologies are implemented individually it can increase divergences between different departments and consequentially the difficulty of merging their information and documentation all together.

Nowadays there are countless EDMS's in the market and they are, usually, associated with a core set of features that may vary slightly from tool to tool and their target business. Some of these features were studied during the development of this thesis, for example: cloud access, document input indexing, searching and processing, version control, workflow automation or user permissions. This type of system contributes directly to the dematerialisation presented in the previous section: automated workflows allow automation on request processing and all the document management features ease the distribution of information in the form of pdf documents.

3. ePaper - System Description

Since the work presented is built on top of *ePaper* and in order to ease the understanding of the topics stated, it is relevant to clarify the meaning of some definitions used on the existing work that will be adopted in the current report. These concepts are related to the direct application of this project, namely the Municipality management of citizen's needs in the topic of urbanism.

- **Process:** is a conceptual entity that gathers all the information regarding an applicant and all his requests. Generally it is created after the first delivery of a form and it usually follows a certain set of phases until its conclusion.
- **Request:** type of application that citizens can ask the municipality. Each request has, typically, one or more forms associated to it and this can be consulted and modified by users. These Requests are listed on the Front Office's public interface for citizens to download ZIP packages with all the needed documentation.
- **Document:** is a digital file (usually a PDF form, but may be an instruction static document or legislation) that has meta data associated to it. Some forms can originate new requests and others are used as complementary information for existing ones. Documents are created by the Municipality and provided online in the Front Office's list of requests interface. After being filled, documents are uploaded to the Front Office platform and they start being read and processed by all the Back Office users until the process is closed.
- **Package:** ZIP file available on the public interface (one for each request), and it contains all the documents a citizen needs to start a process.
  Each package is necessarily associated to a request and contains the document associated to it. Packages are currently created manually by gathering a set of documents into a Zipped folder.

3.1. Users and Use Cases

This section contains a description of some of *ePaper*'s use cases as well of the intervening users.

The set of possible users involved in the whole application process is listed bellow:

- **Municipality Workers**: users who are responsible for handling processes. They review/approve submitted documents and in case of necessity notify applicants of any missing data.
- **Municipality Administrators**: who can manage all system configurations, including documents and request associations as well as package creation and availability.
- **Citizens**: entities who perform requests to the municipality. They download packages on the web platform and submit the filled forms.

Differences between Municipality workers and administrators are shallow and differ from Municipality to Municipality. In practise, administrators have more privileges inside the application due to their job description and, therefore, are able to access a wider set of configurations but a single person might have both roles in a smaller Municipality with less employees.

Follows a brief description of the most relevant use cases:

- **Download Forms** - Citizens access *ePaper*'s Front Office interface and download all the documents they need to start a Request. There it can be downloaded only the initial form or the ZIP package with more documentation,
- **Submit filled documents** - Is the process of actually starting an application that usually is associated with the submission of the filled initial form,
- **Managing Forms Download availability** - System Administrators are responsible for managing available files in the Front Office public interface. This means they create (manually) the ZIP files to be downloaded and they update documents,
- **Create/Set Up Requests** - Documents can only be submitted to a set of typified requests. Those

requests types have to be created *a priori* by the municipality.

- **Process Submitted Requests** - this use case would include an enormous set of possible tasks depending on the type of request made. In any case, for the purpose of this report, it is only relevant to know this whole processing exist.

A typical scenario of *ePaper's* use starts with a Municipality Administrator configuring a Request and associating a Document to it. Afterwards, the Administrator can make the document available on the online public platform where citizens may download it. After filling the document, the citizens upload it and through that action, origin a request. Later on, this request will be processed by Municipality Workers and will follow a determined set of stages until an answer is given back to the citizen with the outcome of the Request.

3.2. *ePaper*'s Structure

The application is divided in two main components: the Front Office which is the public interface citizens should access when they desire to make a request, and another which is the Back Office that works as a platform for users (municipality employees) to manage documents, requests and their availability on the Front Office web page among many other functionalities not necessary for the understanding of this work. Front Office module is independent from Back Office module despite the fact they both access the same database server. They might be installed in different servers which increases the system's complexity once they do not share the same local storage/paths.

3.3. *ePaper*'s Limitations - Problem Analysis

In theory, all human tasks have an associated failure probability and that should be a main concern when designing software and workflows. Not only the way workflows are designed but also the number of validation tasks they contain can have a severe impact on the number of human errors that actually affect a system. This section describes in detail the problems studied.

Reijers et al. [10], associates dysfunctional workflows with some typical characteristics:

- **Data redundancy**, that happens in ePaper, considering that every PDF and ZIP files have to be replicated between Front Office and Back Office Servers.
- **Conflicts between different servers**. The system presents inconsistencies if the files in one server are not exactly the same as the files on the other.
- **Long throughput times**. On tasks such as documents and packages' provisioning for download, given the number of steps it takes to conclude these actions.

All these points can justify the redesign of this workflow, but to confirm this, and to get a better understanding on how to approach the problem, the specifications of the problematic tasks were analysed in detail.

Currently, *ePaper* is not prepared to support some tasks related with document management and package deployment. Despite this, there are a few ways to go around the existing implementation and execute operations that the system is not designed to cover. For example, there is nothing in the system that allows forms to be updated but, as forms are only stored in the file system, if one replaces a form for another totally different with the same name in the same path, *ePaper* will not distinguish between files and will not know the file was changed, but in practise the form is a new file. The lack of rules and procedure validations brings to the system a huge vulnerability to some of the errors described later on this section.

Evidences from customer complaints and users observation led Mind to the belief that the significant majority of the human errors associated with this kind of workflows is directly related with the execution of tasks inside Windows File System.

4. eFormManager

The solution built in this work is called *eFormManager*. This section contains a detailed description of its development from the planning process and objectives definition to its implementation technicalities.

4.1. Objectives

In order to clarify this work's requirements, Wiegers' [11] approach was followed. Several types of requirements are to be defined in the beginning of the work and will be used as guidelines as the project goes forward. A generic scope of the project should be defined as well as what should be achieved with the project's outcome. The vision statement, following a Wiegers' template helps clarifying all the generic aspects of the project:

Most Portuguese Municipalities currently provide online all the forms and documentation their citizens might need to perform any type of requests. Managing all the documentation and its provisioning is executed manually at present, without any type of support tool or validation, which is problematic when updating and distributing hundreds of forms. The product presented in this report is a document and request manager that aims to ease the delivery of sets of versioned documents through automation of tasks, a clear interface and error detection features. Its major benefits are the validation of human actions, automated control of versioned documents, data coherency when fetching up-to-date documentation and mostly the automation of document/packages delivery. Unlike typical document managers whose focus is on organising and presenting different versions of documents, this application provides some of their functionalities having the extra possibility of scheduling deliveries and gathering documents in sets to be automatically delivered.

4.2. Solution Requirements

After an agreement on the project's generic objectives, users and use cases were studied in order to define functional and non-functional requirements.

4.3. User Classes and Use Conditions

There will be two different types of users which will use the application with different frequencies and under different circumstances:

- **Municipality system's Administrators** who perform all the management of the application. Usually there are very few administrators per municipality so it is unlikely that the tool is being used by two users simultaneously.
- **Citizens** that can access the public platform and download packages - the number of downloads can be considerably big (may occasionally reach hundreds per day)

Besides this, an analysis was made on the use of the currently installed versions of the *ePaper* application, and how often these various types of actions are performed.

Usually, on the first configuration of the application, administrators create all the requests, assign them the forms needed to apply and make them available. After that, documents and packages are changed sporadically (when legislation or internal processes change, for example) meaning its capacity to scale in terms of simultaneous user accesses is negligible. On the other hand, *ePaper*'s Back Office application is not accessible from the outside of Municipalities once it runs on their internal networks and, therefore, capacity to answer to diverse requests is not a concern of this tool but instead of the servers that run the Front Office public platform.

4.3.1 Functional Requirements

In order for functional requirements to work as evaluation criteria for the work developed, they should follow the SMART requirements [12] designation being Specific, Measurable, Attainable, Realisable and Traceable. With functional requirements well defined, it is possible to evaluate objectively the final product in what concerns its features. A list of functional requirements is presented here:

- Document insertion - the system should accept files (typically PDFs) and store them for download and management.
- Request creation - the system should allow the configuration of requests citizens may apply for.
- Association between Requests and Documents - the system should have configurable sets of documents associated with each request that need to be delivered in order to perform the respective request.
- Document update - the system should allow to add files as documents' new versions.
- Document version schedule - the system should have a date for every document version from which it becomes active and is used instead of previous ones.
- Version comparison - the system should be able to compare different versions of a PDF form, concerning its input fields.
- Input testing/validation - the system should detect and warn against "potentially" wrong inputs (files, dates, text inputs, etc). For example, if two files are totally different and one is inserted to replace the other as a new version, probably either the selected file was wrong or the document being updated is not the one desired and the user should be warned (avoiding as much as possible the human errors).
- Submitted files' validation - the system should verify whether a file corresponds to the currently active version.
- Download of documents and packages - the system should allow the download of documents and packages for every request providing automatically the up-to-date version of every document.
- Automatic package creation - the system should automatically provide packages for every Request that have the active versions of the all associated documents
- Activities' logging - keep logs of all the changes made to each Request and each document.

4.3.2 Non-Functional Requirements

Nonfunctional requirements are described as characteristics the system must exhibit or properties it must have. The concrete goal of *eFormManager* is to be integrated with the *ePaper* application which runs in Municipalities' internal networks. Therefore, all its security requirements (such as users, logins or permissions) will be inherited from *ePaper* after its integration.Besides this, as explained before, the new tool is not going to have more than half a dozen users in total and therefore, server capacity to deal with many accesses or even simultaneous requests is not one of its main priorities either.

The non functional requirements set for this project are listed bellow:

- **N.F.R.1** - Performance - users should be able to complete tasks like creating packages or updating documents in less time and with a fewer number of errors than they were before.
- **N.F.R.2** - Accessibility - for this application to be used in public administration, it has to follow accessibility rules issued by the Portuguese Government in [13]. A verification tool [14], also provided by the government will verify that the system's interface complies with all the rules.
- **N.F.R.3** - Usability - can be expressed by the complexity of the actions needed to perform a task or the number of clicks needed to conclude the desired task. The usability requirement in this context is to have a tool that is easier and more intuitive to use than the previous system and it should be reflected in user satisfaction.
- **N.F.R.4** - Openness - even though it is built to be integrated with *ePaper*, this tool is supposed to be able to be integrated with other systems. Therefore, its architecture has to be ready for integration

with other applications.

All the requirements referred in the present section will be covered again in section 5.

4.4. Use Cases

In order to introduce the tool developed, its use cases are described briefly.

- **Create Requests**: the creation of a request entity, that can be followed by an association with existing Documents.
- **Insertion of new Documents**: inserting in the system a new PDF file that will be read and can afterwards be associated with any existing Requests.
- **Update Documents**: consists on inserting a new PDF file on the system that will be considered the most recent version of the current Document. When a new version is inserted, the user has to choose a date from which the version becomes active (downloadable).
- **Compare Versions**: a user can at any time select two versions of a document and have a visual comparison of the PDF forms selected where the different fields of the forms are highlighted. This comparison is also made whenever a new version is to be inserted so that the user can verify the correctness of his action.
- **Consult Log History**: Admins can consult the actions taken into each request and the history of versions of each document.
- **Package Download**: The application has an interface (and an API) for package download. When an applicant downloads a request, the system verifies the package is up-to-date and creates a new one in the case the current package is out-dated.

4.5. Solution Architecture

The *eFormManager* Application's structure is illustrated by the diagram of figure 1. Contrarily to *ePaper* , the *eFormManager* is designed to run in a single server, having all the business logic and file system in the same place which increases data coherency.

One of the components of the system is a C# library where all the logic is encapsulated and divided in multiple modules as shown in the diagram above. These modules correspond to classes created in order to separate the code of different functionalities. Apart from this, there is a controller where web requests are received and forwarded to be processed in the logic modules. There is also a component of web pages and scripts that make the User interface module and a data layer that provides an interface to access the database.

4.6. Architectural Models

It is true that interfaces and styles have to be frequently remade and redesigned, but the same does not apply with the logic behind the program. Therefore, building a RESTful application provides total independence between front-end and back-end code. This architectural style implies that the back-end has a well defined set of

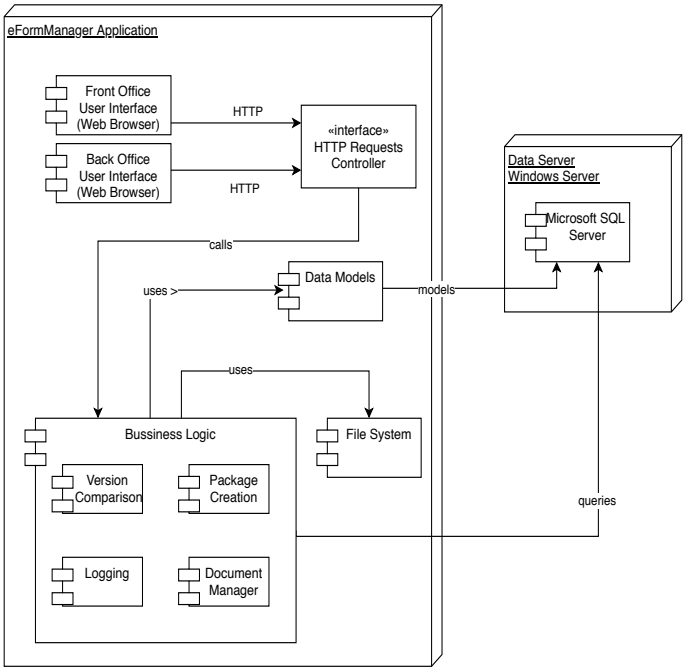


Figure 1: eFormManager Architecture

endpoints to be called for different tasks/services and these endpoints are the ones who may facilitate an integration with any system. Considering this tool is to be integrated in another system, it is useful to have a SPA (single page application). Besides, SPA's and REST API's are a good combination to have: information fetched from the server in JSON format can be processed by the client and displayed asynchronously without reloading the page. No framework was used to develop the client-side application. Instead, some JavaScript design patterns [15] were analysed and an MVC (model, view, controller) design pattern was chosen for being a good match to SPA's.

5. System Evaluation

In order to evaluate the system, should be asserted whether all the predefined requirements were matched or not. Validating functional requirements should be done for technicians and managers who evaluate functionalities compliance with what was set and can be demonstrated through screen shots which would be out of the scope of this article. Inspite of this, non functional requirements will be analysed in this section.

5.1. Performance Evaluation (N.F.R.1)

eFormManager is not to be used by more than one user simultaneously which makes load tests less relevant once there are not more than 3/4 users per municipality with privileges to access it.

There is a small part of *eFormManager* that is available outside of the municipalities intranet. Although, it would not make sense to test its capacity to deal with huge loads of requests once all those requests are made to the Municipality server (which is out of the scope of this project) and it will use the *url* provided by the application for a request. Given this, the application only executes work when a document or request is updated (which does not happen often enough to be

relevant) and, therefore, the workload is on the municipality server and not on *eFormManager*.

To obtain some performance metrics, Google Lighthouse[1] was used and a report was generated. According to Lighthouse scoring measures, tests had a score of 100/100. These tests were executed with *eFormManager* running on the local-host which will not be much different from the reality once it will always be ran in internal networks. The data volume used to execute the tests was a database collected from the most populated municipality that uses ePaper, so it can be considered a real case and still be working with a large data volume.

5.2. Accessibility Evaluation (N.F.R.2)

Concerning the human-computer interaction, accessibility refers to the use of software by people with some kind of disability. Systems should be built with some special characteristics in order to be understood and read by assistive technology software used by disabled users such as screen readers that read HTML pages, in the case of blind people. These software tools have limitations and the pages they read must be compliant with a set of rules.

Following an European Union's directive [16], in October 2018 , the Portuguese Government emitted a Law [13] with a set of requirements that every web site of a public institution must comply with. To help verifying this, a tool [14] was provided online for developers to insert their files or URl's and a score is given in a scale of 1 to 10 regarding all the rules set in the legislation.

Besides this institutional validator, other development helping tools evaluated the application such as Lighthouse Audit tool used by Google Chrome. With Lighthouse, a perfect validation of all accessibility rules was obtained.

With all these validations, accessibility was considered a matched requirement.

5.3. Usability Evaluation (N.F.R.3)

Usability is a crucial parameter for measuring the success of this work, once it aims to improve user experience in a set of workflows as well as to decrease the number of errors associated to them. This section quantifies the improvement that this system brings in comparison to the previous workflows. As mentioned in section 4.3.2, usability can only be measured through user experiences and feedback and therefore tests were performed with real users.

5.3.1 Testing Set Up

There are a few standardised software usability tests that quantify the quality of users' experience. In the context of this work, two different tests were chosen for two different reasons.

- **System Usability Scale (SUS)** [18]: introduced by Brooke in 1996, was chosen to classify globally only the system built and also the previous system and provide a comparison based on the same criteria.
- **After-Scenario Questionnaire (ASQ)**[19]: based on Lewis, J. R. (1995) , it consists on having a small set of threes questions after the execution of a guided task. This method was considered extremely useful for the purpose of comparing workflows and specific scenarios. If a tester executes the same task with both tools and answers the same set of questions rating the tools within the same scale, that would be an explicit measure of comparison.

For the execution of both tests, an ideal scenario would be to have users experimenting all the features of both systems having a full experience and giving feedback afterwards. Having this in consideration, three tasks were chosen and described in a guide so that the tester would go through the vast majority of the implemented features.

These tasks are related with the management and deployment of forms and packages and are representative of a real world scenario: i) the creation of a ZIP package correspondent to a Request with a form inside and its deployment for download; ii) adding two auxiliary forms to the previously created package and its deploy for download; iii) the change/update of the main form and the deployment of an up-to-date package (with the form's most recent version).

Ideally, the target population would be composed by real users of the software and potential future users, i.e municipalities administrators. Unfortunately, for commercial reasons that was not a possibility during the execution time of this project. Despite this, an effort was made to build a profile of the real user and some characteristics were found relevant for testers to have.

On the one hand, it is relevant to have testers with good knowledge of *ePaper*'s workflows once they are the ones that can actually measure the improvement brought by new system. This subset of the population could only be found inside the company and so, 10 members of the Support and Testing teams were asked to participate in the evaluation. On the other hand, for the purpose of this work, a non-biased comparison between workflows is a good qualifier of the work done and users that have never used *ePaper* should give a relevant input for this study as well as they can represent potential new users' behaviour. In order to limit the universe of testers and consequentially have data as close to reality as possible, the typical set of skills of the ultimate users was studied. A set of 10 testers that didn't know *ePaper* was collected making a total of 20 testers.

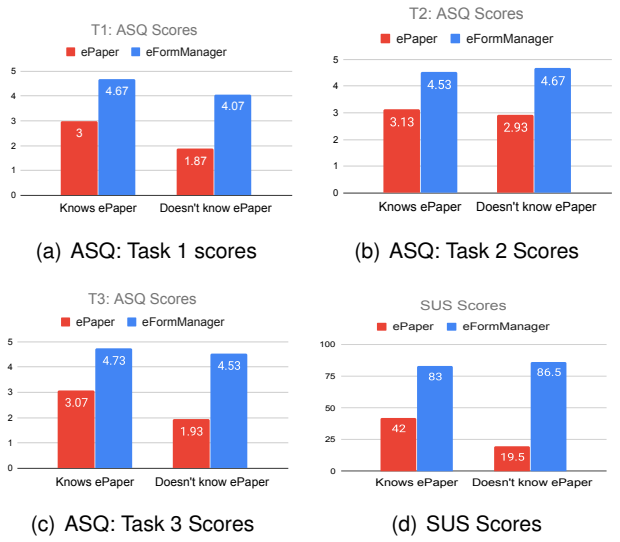
5.3.2 Results and Analysis

Given the variety of tests executed and results obtained, different analysis had to be made. Despite this, it can be concluded for each and every test that the results were very satisfying.

The results (presented in Figure 2) can be majorly

[1] https://developers.google.com/web/tools/lighthouse/

divided in two subgroups: the testers that had prior knowledge of *ePaper* and the ones having their first contact with both systems. Obviously, being this a comparison between two systems, it is extremely relevant to know whether testers knew *ePaper* in advance or not. As expected, results are quite distinct for these two groups and it is worth to analyse them separately.



(a) ASQ: Task 1 scores (b) ASQ: Task 2 Scores

(c) ASQ: Task 3 Scores (d) SUS Scores

Figure 2: Usability Tests scores

5.3.3 After-Scenario Questionnaire

Every question of each task is answered with a score from 1-5 regarding how easy it is to complete the task, the time spent, and the support information were given.

Task 1 results are illustrated in figure 2(a). It is the task where the results for the *eFormManager* are lower. This could be expected because, in fact, this is the most difficult task to execute - the extra complexity on this task allows for decreasing all the complexity on the remaining workflows. Despite this, there is a substantial improvement when comparing *ePaper* (3, 1.87) with *eFormManager* (4.67, 4.07) for both populations of testers. It is interesting to highlight that users who know the complexity of existing functionalities gave a significantly higher score to the new system.

Task 2 (Figure 2(b)) is the task where there is less discrepancy between both systems. This does not mean that the results are negative, once scores for the *eFormManager* are above 4.5 out of 5 and it can be explained by the following reason. The execution of this task in the *ePaper* application consisted only in the creation of a ZIP file in the File System and its renaming which is in fact quite simple. For this task, major improvements should be observed when analysing the number of errors occurred in its execution.

Task 3 has very satisfying results for *eFormManager* (4.73 , 4.53) (Figure 2(c)) opposing to *ePaper*'s (3.07, 1.93) and it is where there can be observed the biggest improvement in terms of classification. When compared to *ePaper*, the new system has some significant differences, namely in the automation of package delivery and it justifies the user satisfaction.

5.3.4 System Usability Scale

Results of SUS tests are shown in Figure 2(d). For both groups of testers, results show a better score for the *eFormManager* which is the most relevant fact. As predicted, the SUS score for *ePaper* is not satisfying and, even though its absolute value almost doubles for users that are familiar with it, both scores correspond to an F on the benchmark used[20]. For the *eFormManager*, scores are quite similar for both groups and they float between an A^+ and an A grade.

Scores of 83 and 86.5 are extremely satisfying scores for the usability of *eFormManager*, mostly when compared with the scores of *ePaper* which are considered poor in this tasks.

It is understandable that users that knew *ePaper* in advance are more aware of its workflows and therefore gave it a better score than users who did not know *ePaper.* But this only increases the relevance of the improvement noticed when comparing scores from ePaper and eFormManager.

Regarding the usability of the *eFormManager*, there is a clear improvement considering the previous implementation of such functionalities and, therefore, it can be said the objective was achieved.

5.3.5 Task Completion Times

The time users spent completing the tasks can be considered a relevant measure for the users facility to complete tasks. The average amount of time used by users to execute all tasks is presented in figure 3. Obviously, there are many factors that may influence the time taken by a user to complete the tasks and that will be mentioned in this section.

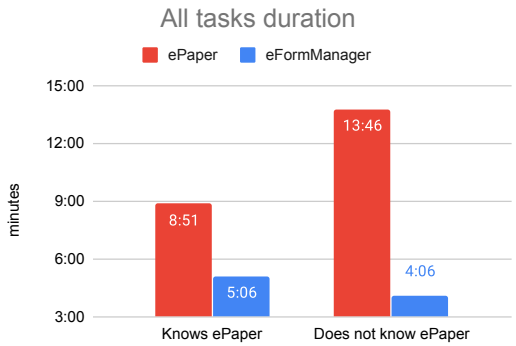


Figure 3: Tasks duration Time

At first sight, can be observed a big discrepancy in the average execution time which was smaller for *eFormManager* showing a positive result. Curiously, though, *ePaper* users spent more time using *eFormManager* than new users. During test execution, *ePaper* users had more comments and questions towards the new system, its different workflows and some conceptual changes and that can justify the extra time they spent. As new users were not biased into any workflow and the discrepancy between the times they spent in

each system is remarkable, this is an extremely satisfying result.

5.3.6 Errors occurred

During test execution, an observer noted down the errors users made when following the guide. On purpose, users were only notified of their errors when they led to a point in which the user could not understand why their actions did not give the expected output.

The following table displays the number of errors that occurred during the execution of the tests in both systems. In order to simplify its analysis, the errors were aggregated in 4 categories:

- Wrong ZIP creation: this error category includes all the errors associated with the creation of a ZIP file, namely, the forms that it contains and the folder where it was placed.
- Wrong ZIP naming: even though, the name of the ZIP is directly related with its creation, this was a very common error and so it is displayed in a different category.
- Wrong File Selection: when through the interface users have to pick a file (in a dialog box or drop down list).
- Wrong Form Input: when users inserted wrong information in the system through miss clicking or writing mistakes.

Table 1 shows the number of errors by type that happened during the tests. The *eFormManager* presents a significant improvement regarding error occurrence. The last line of the table has the number of errors the system detected and consequentially prompted the user.

The fields in the table marked with "-" correspond to non applicable situations, for example, *eFormManager* not having any contact with ZIP files.

Error Type	ePaper			eFormManager		
	T1	T2	T3	T1	T2	T3
Wrong ZIP Creation	5	8	15	-	-	-
Wrong ZIP Naming	8	10	9	-	-	-
Wrong File Selection	7	-	0	0	0	3
Wrong Form Input	3	-	3	2	0	0
Total	68			5		
Warnings Prompted	12			5		

Table 1: Number of errors in each task per system

As Table 1 shows, ePaper usage is associated with many human errors (which in a way is a validation of the problem that led to this work) while, in that matter, the solution developed presents way more satisfying results.

Another relevant data from this results is the number of errors that each system detected. Not only *eFormManager* had fewer errors but also all the errors were detected by the system. The version submission validation feature was extremely helpful in this topic since the most common error was users choosing the wrong file as new version and the verification performed alerted the users for a potentially wrong insertion allowing a clear confirmation.

On the other hand, ePaper has an enormous percentage of undetected errors once it is not ready to verify the operations executed exclusively inside Windows File System (where the majority of errors occurred).

The number of errors and their detection is an extremely positive measure of success for *eFormManager* considering that one of the initial goals was to increase control and change error related workflows making them less vulnerable.

5.4. Openness Evaluation (N.F.R.4)

A qualitative evaluation can be performed regarding how easy it would be to integrate *eFormManager* with other systems. As described in section 4.5, REST architecture provides some characteristics to the system built that were considered to be crucial in order to ease the integration of *eFormManager* in other systems.

The key point about this architecture is that it provides a clear interface of endpoints that can be used not only by JavaScript Web Browser clients but also by any other entity with the right privileges that has access to those endpoints.

This means that another system with its own interface can make HTTP requests to this system and perform actions in its database as well as fetch data or files from it. Considering this, REST endpoints can be used as an API that provides undoubtedly a huge degree of openness to this system.

If we consider the specific example of *ePaper* and its two major modules, *eFormManager* should be fully integrated in the Back Office module but it should easily be called from the Front Office module through HTTP Requests in order to fetch the organisational tree with the requests and also the packages for download.

6. Conclusions

Public administrations are constantly increasing their investment in tools that automate document management related jobs, not only tasks directly related with files themselves, but also their distribution for the population and the following reception. Online and automated help desks are becoming common and the features they present are growing in complexity and variety.

The application developed in this work - *eFormManager* - aims to increase the automation of Portuguese Municipalities' documents life cycle since they are created until they are deployed for citizens to submit them online. This process became more efficient by reducing the human intervention needed and supervising the remaining human action through algorithms that validate tasks executed by users prompting them if potential errors are detected.

The work was developed on top of concepts and paradigms of the *ePaper* system by Mind on which the application built is to be integrated. *ePaper* is used for urban management by Portuguese Municipalities having tools for PDF form management and their distribu-

tion to the population who can also use *ePaper*'s public platform to submit their filled requests.

Even though the final goal of this work is to integrate *eFormManager* into Mind's application *ePaper*, an independent web tool was built, tested and evaluated. Since this tool is to be integrated with another system, it was built with an architecture that provides isolation between the front-end, the back-end and its database so that each one of these components could be re-used independently.

Despite being built within the context of the *ePaper* application and having certain requirements of features and specifications to follow, the tool presents some extra features such as document comparison and changes' tracking . These features are described throughout this report and increase its value beyond objectives initially defined.

In order to evaluate this tool (its features and interfaces), an evaluation survey was performed where 20 participants compared the use of the new tool with the previous workflows (in *ePaper*). Statistical results were extremely satisfying since usability and user workflows results are indicative of significant improvements. The following results should be highlighted:

- User satisfaction results were clearly better for *eFormManager* than *ePaper*,
- The time spent on the execution of the same tasks was smaller when using *eFormManager*,
- The number of errors occurred during the execution of the tests was extremely smaller for *eFormManager* when compared with *ePaper*.

Besides, the evaluation survey was very relevant because of the inputs received from the participants and led to some corrections and improvements of the application.

References

[1] *Resolução do Conselho de Ministros n.º 51/2017*, Diário da República n.º 77/2017, Série I de 2017-04-19. Accessed 13 September 2020. Available at https://dre.pt/home/-/dre/106896737/details/maximized

[2] C.T. Goodsell, *The public encounter: Where state and citizen meet*, Ind. Un. Press, 1981, pp.3-20

[3] Lindgren, I., Madsen, C.Ø., Hofmann, S., Melin, U., *Close encounters of the digital kind: a research agenda for the digitalization of public services*, 2019, Gov. Inf. Quarterly, Vol. 36, 2019

[4] Armenia, S. & Canini, Daniele & Casalino, Nunzio. (2007). *A system dynamics approach to the Paper Dematerialization Process in the Italian Public Administration.* The Interdisciplinary Aspects of Information Systems Studies, D'Atri A., De Marco M., Casalino N. (Eds),Physica-Verlag, Springer, Heidelberg, Germany, pp. 399-408, 2008

[5] Casalino, Nunzio & Armenia, S. & Draoli, Mauro. *A System Dynamics model to identify and measure the paper digitization advantages in Public Administration.* .Management of the Interconnected World,, Physica/Springer-Verlag, 2009

[6] Ford, A. *Modeling the Environment: An Introduction to System Dynamics Modeling of Environmental Systems*, Int. J. of Sustainability in Higher Ed., (2000),Vol. 1 No. 1.

[7] Ostroukh, A. V., Krasnyanskiy, M. N., Karpushkin, S. V., & Obukhov, A. D. *Development of Automated Control System for University Research Projects*. Middle-East Journal of Scientific Research, vol.20, no.12, pp. 1780-1784, 2014

[8] Sprague, Ralph H., *Electronic Document Management: Challenges and Opportunities for Information Systems Managers*." MIS Quarterly, vol. 19, no. 1, 1995, pp. 29–49.

[9] Haider, A.A., Aryati, B. and Mahadi, B. *Opportunities and challenges in implementing Electronic Document* Management Systems. Asian Journal of Applied Sciences, 2015, 36–39.

[10] Reijers, Hajo & Mansar, Selma & Aalst, Wil M. P.. (2003). *Product- Based Workflow Design*. J. of Management Information Systems. 20. 229-262.

[11] Wiegers, Karl & Beatty, Joy. (2013). *Software Requirements Third Edition*. Microsoft Press, Redmond, WA, USA.

[12] Mannion, Mike & Keepence, Barry.(1995). *SMART Requirements*.Software Engineering Research Group.Napier University

[13] *Decreto-Lei n.º 83/2018* Diário da República n.º 202/2018, Série I de 2018-10-19, Accessed 13 September 2020. Available at https://dre.pt/home/-/dre/116734769/details/maximized

[14] Fundação para a Ciência e a Tecnologia, IP - Unidade acesso da FCT - Accessed 30 May 2020. Available at *http://accessmonitor.acessibilidade.gov.pt/amp/*

[15] Addy Osmani, (2017) *Learning JavaScript Design Patterns*, O'Reilly

[16] *Directive (EU) 2016/2102 of the European Parliament and of the Council of 26 October 2016* on the accessibility of the websites and mobile applications of public sector bodies

[17] Web Content Accessibility Guidelines 2.0,(2008) W3C World Wide Web Consortium Recommendation *at http://www.w3.org/TR/WCAG20/*

[18] *SUS-A quick and dirty usability scale* (in "Usability Evaluation in Industry", PW Jordan, B Thomas, I McLelland, BA Weerdmeester (eds)), John Brooke, 1996

[19] James R. Lewis. 1995. *IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use*. Int. J. Hum.-Comput. Interact. 7, 1 (January 1995)

[20] Lewis, James & Sauro, Jeff. (2018). *Item Benchmarks for the System Usability Scale.* J Usability Stud. 2018; 13:158–67.