

Whitepaper:

**Testing Document
Management Systems
in the Legal Sector**



Introduction

The document management system (DMS) is a central component of a firm's IT systems. When looking to upgrade, a robust test approach will provide confidence that the new system will work effectively.



Whether a firm is implementing iManage, NetDocuments, Sharepoint or another option, it is rare that the DMS will operate in isolation. There are likely to be many integrations into a firm's other systems. These integrations can be for external systems to store or retrieve documents, to update or create folder structures, or to manage permissions.

Testing of the DMS implementation will look to determine the configurations and customisations have been implemented according to specification. Testing of the integrations between the DMS and external systems seeks to validate other applications will still work as expected once the DMS has changed. Undertaking performance testing and user acceptance testing (UAT) will provide information to understand whether the new user experience will inhibit successful adoption of the new solution.

Planning your DMS testing

Unlike a practice management system (PMS), client onboarding tool, knowledge tool or client relationship management system (CRM), there is no obvious business unit to own the DMS, so it often falls to an IT department. This can have implications on getting decisions regarding the correct behaviour of the system.

The first exercise that will need to be completed as part of the testing of the document management system is to understand:

1. All the integration points
2. The custom behaviours that are being implemented within the DMS software
3. The environments the system will be



presented from and the types of users accessing it

4. The data migration requirements

This test coverage will need to include all the integrations of the tool, and the firm specific customisations. It is often the case that integrated applications themselves need to be upgraded in step with the DMS further widening the scope of the testing.

Existing test collateral, such as test scripts may already be in place which will no doubt be a good accelerator. However, it is likely that significant changes will need to be made to the scripts to incorporate new interface designs, updated work flows and new requirements. If no existing test scripts exist these will need to be created.

Scripts should ideally be stored within a test management tool, this will help with structuring the scripts in a way that allows different suites to be run in different circumstances. By taking this approach, you will improve long-term efficiency, for example, it may be the case that an update is made to an environment that only requires



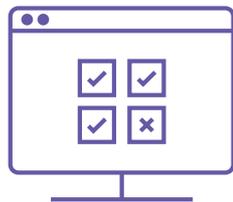
123rf/ dolgachov

a specific set of tests to be executed. If the scripts have been organised well it will be easy to identify these scripts, reducing the effort required in execution.

Functional testing

With the analysis of what needs to be tested complete, the execution of these tests can begin. To execute the tests, the environmental dependencies will need to be in place. It is likely that testing needs to be completed against

different platforms e.g. laptop build, desktop build, Citrix. There may be different user account types that need to be simulated such as a fee-earner or PA. Test workspaces with test documents will need to be created for each of these user personas.



The data within test workspaces should be as representative as possible, with document security, size, complexity, variety and volumes all being considered. It's at this stage that the security and permission model that is being applied within the firm will make a big impact to the tester. Firms will usually opt for an optimistic or pessimistic security model. Historically, most firm's have employed an optimistic model where specific workspaces or documents have had restrictions applied to them to limit who can view them. More recently, firms have been employing pessimistic models where documents and workspaces are restricted by default and mechanisms are put in place to grant access to only those that have a requirement to view the information. To achieve a representative test data set this security model will make a big difference. The crafting of this test data can take a lot of effort but it is imperative to making the testing comprehensive.

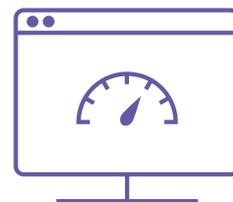
Execution of your testing can then follow normal workflows. With so many variables involved with the data, it is important to follow best practice when raising defects and to contain as much information as possible to enable the technical team to recreate the issue. By including enough environment information the fixes for defects can be created quickly and efficiently.

The functional testing can only provide so much confidence that a DMS implementation will go smoothly for the end user. It is advisable to undertake other testing to boost this confidence.

Performance testing

Performance testing of the new DMS will look to provide metrics around the user experience. Where possible, it will be helpful if you can benchmark your new performance testing metrics against the existing DMS to determine whether there has been a performance gain or regression. It is important that only like-for-like transactions are compared though. Should it not

be possible to gather metrics for the existing system, performance testing can still provide a good model to help with user engagement and also as a baseline for future implementations.



The performance of several interactions should be considered, and for each of these interactions multiple parameters can impact the performance. Examples of interactions include:

1. Launch the main client interfaces of the DMS
2. Check in/check out a document
3. Perform a search
4. Navigate around a library

Parameters that may need to be considered include:

1. Document size
2. Document type
3. Number of documents in a workspace
4. Global location of user
5. User load

It is becoming more common for the server-side components of the DMS to be cloud based rather than on premises. This new architecture can have a significant impact on performance. It may be the case, particularly for a global firm, that the physical distance between the user and their documents increases. This distance introduces additional latency and can significantly impact the user experience. To be able to manage expectations on the users as part of user engagement it is important to fully understand where things may be faster and where things may be slower.

Performance testing can also be used to help calibrate advanced monitoring systems. These monitoring systems can provide a lot of data that is abstracted from the user experience. By investing in performance testing, it is possible to correlate how the monitoring systems will report on specific changes to the user experience.

It is always important to conduct performance testing against a representative environment. If future performance testing is to be planned then there needs to be a plan to retain a production-scaled, non-production environment.



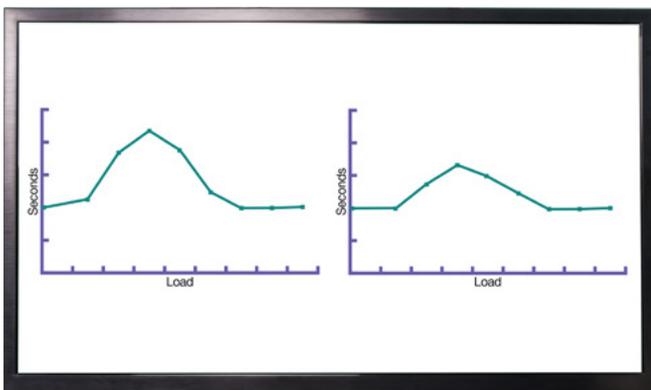
123rf/ Andor Bujdosó

User acceptance testing (UAT)

User acceptance testing is an opportunity to get the new DMS in front of users as early as possible. The purpose of this phase of testing is to ensure that the requirements captured prior to implementation were correct and complete. By having users interact with the application as early as possible, important feedback will be captured that can assist in a successful go-live. The environment used for UAT is an important consideration. If users are going to be using the upgraded system for real work then the integrity of the data must be guaranteed.

Feedback from UAT can be useful to highlight several different types of problems. Defects that may have been missed as part of functional testing may be highlighted and issues caused by missed requirements may be identified.

Comments around the user experience should be captured, especially those around performance. Training, floor walking and early life support can provide observations of when a user finds it difficult to understand how to do something.



To maximise the benefit from a UAT phase, there needs to be good engagement with the user identified as part of the UAT group. The test team needs to make it as easy and non-intrusive as possible for the users to complete their testing to maximise response rate. Potentially low response rates mean it may be advisable to have multiple users covering each role to still ensure full coverage of UAT scenarios.

Conclusion

Having gone through the aforementioned test phases you can be confident that:

- the DMS configurations have been implemented as intended
- external applications dependent on the DMS will still work as expected
- the solution is performing to the expectations of the users
- the new implementation is able to fulfil the needs of the users

Alongside effective user engagement, training and quality early life support the go-live should be a success.

The successful go live of an upgraded DMS is not the end of the testing. As the external applications that integrate with the DMS change then regression testing may need to take place.

The platforms that host the server components of the DMS or the clients used to access the DMS will need frequent patching meaning the DMS should be tested. As a firm's requirements of its DMS changes then there may need to be updates to the DMS which will need testing. By implementing a solid foundation of comprehensive, robust and well written test scripts this future testing should be as efficient as possible.

Is this now time to consider test automation? Automation of a DMS regression pack should enable changes to any part of the infrastructure to be implemented quickly and with the confidence users will still have the necessary access to the documents they require to fulfil their roles.