

DOCUMENT AUTOMATION CHECKLIST 2020

What is Document Automation?

Document Automation is the process in which one answers a set of questions in order to generate a tailored legal document from a standardized template. When discussing a transaction with a client, lawyers are often thinking about how to document certain items in the contract. With document automation, most pathways dealing with commonly occurring issues have been codified into the document template, meaning groups of words can be switched in and out with a simple press of a button.

As a result, document automation solutions can offer substantial efficiency gains, as drafting time can be cut as much as 90%. For example, a set of documents that used to take eight hours to prepare manually can be done in less than an hour. Document automation also reduces simple but common errors, such as forgetting to change a related concept from one place or document into another.

What is Computer Assisted Legal Drafting?

Computer Assisted Legal Drafting takes document automation to the next level by adding the key feature of flexibility. This flexibility comes from the option for freestyle editing, which is a free-flowing editing ability that can be turned on or off, allowing the user to edit any portion of the document. Freestyle editing also reduces decision fatigue associated from traditional document automation driven from the numerous questions and sub-questions that must be made regarding language in the document.

Computer assisted legal drafting has several benefits, first of which is allowing attorneys to devise new or custom language to precisely effectuate the desired terms without being limited to multiple choice options. Second, each transaction is securely recorded in the automation system, making it easier to complete similar deals with the same or different party later on. Third, there are accurate, consistent, and compliant guardrails in place, no matter who is drafting.

This innovation is technically difficult to achieve on a general level automation system, and currently only Celant provides a document automation solution with the freestyle edit feature. That said, if you want to evaluate or review the various document automation solutions on the market today – what should you look for?

Following are a series of features to look for, and questions to ask, to help you effectively evaluate document automation software to find the right fit for your organization's needs.

Your document automation solution should:

1. **Be web-based.** Why? So you don't have to manage installation, and more importantly, maintenance, on each of the desktop computers. Web is the ideal method of delivery for modern software, and it's easy to share work-in-progress with your colleagues in that medium, so that team members can easily contribute to the work product. Although there are certain security benefits to being removed from the web, for most collaborative business applications that are actively maintained, the web-based platforms tend to outclass isolated programs.

2. **Be a stand-alone app, not an add-on.** If it is a MS Word Add-on, the automation software needs to be updated for each update of the underlying MS Word installation. To overhaul every few years just to maintain functionality is extremely resource draining for all parties involved, and it requires maintenance of multiple versions which dilutes resources for the developer. You might not care about the add-on software company's problems, but it will inevitably show up in the price. It may also affect your company's decision on when to upgrade MS Office or even Windows version, due to lack of upgradeable versions in the legacy software.
3. **Dynamically auto-generate interview questions.** The modern document automation solution should be able to tell whether a field is used or not in the document, and based on its use (or potential use) based on user's choices, limit the questions to only those relevant, without having to manually program this behavior (or arrange variables in certain order), and present it in the order it appears in the document. This saves huge amount of setup time. One benefit of manually programming the interview question is that you get to choose the order of questions; however, this point becomes moot if the input form, term sheet, abstraction or summary is organized in the intuitive order (which they usually are) and presented first because the questions will follow its order.
4. **Modify multiple documents at once.** Transaction documents do not travel alone; they travel in sets. So it only makes sense that if you input your name, it is populated in all relevant documents within the transaction. This means that summary documents, like term sheets or abstractions, and legally binding documents, like merger agreements or land contracts, are created using the same data at the same time, ensuring that they match the information perfectly. If the term sheet, and the actual agreement differ due to intervening negotiations, such differences should be noted as a different version. This way, you *always* have the accurate term sheet or lease abstraction handy.
5. **Show the underlying document in real time.** For consumer facing application, display of actual document does not matter because consumers do not care about the language as much, but for professionals, the ability to clearly see how a choice is being manifested in the document is paramount. It should not require a back-and-forth navigating between the interview and document. Ideally, such changes should be shown in the "track changes" format. This ensures experienced practitioners that the document is being formed correctly, and allows new practitioners to learn how to effectuate certain business concepts onto contract.
6. **Be able to make freestyle text edits.** Freestyle editing refers to the ability to modify any portion of the document, not just where the input fields are located. It is simply not feasible to anticipate requests that might be reasonably made by the non-drafting party. If one were to even attempt it, that would quickly lead to decision fatigue on the user due to numerous questions that must be asked to achieve it; it is better to grant the user the freedom to insert or delete any words in the automation system. If the automation system cannot do it, the less optimal alternative is to convert to Word DOCX, and then edit it in MS Word. Then, you now have two versions in two different platforms to keep track of, a nightmare scenario wherein the efficiency gain of an automation system is essentially vitiating.
7. **Keep track of different versions and report the differences.** Binding agreements are frequently negotiated. Negotiations are done as a package, not piece-meal (you can't simply cherry-pick the best parts). Users must be able to 'save' an older version of the document that was sent to the other side and it must be kept track of. The automation system should be able to handle versioning, instead of having to rely on other systems, and effectively report the differences and communicate to the user, because differences between the versions are best described as differing choices, rather than different words. Utilizing the structured data contained in the system, topical comparisons are possible, so that the "issues list" can be generated from the comparison, in addition to the traditional "redlines" which requires human intervention to be made useful.

8. **Automate workflow of related tasks.** It is not enough to automate the generation of documents; the system must be able to automate mundane tasks relating to document generation, such as integrating with calendar for consultation, keep track of the approval chain, assign personnel based on the current status of the document, and send email alerts or text reminders upon completion of a designated event or an upcoming appointment. When these activities around the document generation are automated, the total productivity gains can be substantially more impactful than just automation of documents alone.
9. **Be easy to maintain the templates.** The forms change over time, so it is important that the templates representing the forms be made easy to update. No computer programming knowledge should be necessary to maintain the forms. Some type of mark-up styling (like HTML) that is intuitive and easy to learn is ideal, but in any case, it should not require a template author to learn new proprietary computer programming language. In addition, it should be able to maintain separate versions of each templates.
10. **Be able to create e-signed documents and closing binders.** The document automation system should be able to reach the finishing line. Creation of the closing set means inserting the signed signature PDFs at the right places, combining the rest to form fully executed documents in PDF, all at once. The way to achieve this may differ based on the system - some systems might integrate with DocuSign or have other intuitive means of doing so. It should be able to accept input of the signatures, and thereafter, it should be able to automatically generate the final closing set or executed documents.
11. **Keep track of material terms of the contract.** After the deal is done, some specifiable and customizable key parameters such as the closing date, the transaction size, or counterparty's name, should be made available outside of the document, so that it can be sorted or filtered based on that information. For deal execution professionals, closing is the end, but for others, it is the starting point. This improves the "handoff" to the professionals handling the next stage of the contract cycle.
12. **Be easy to implement and to customize.** The implementation of a document automation system often requires hiring of outside consultants and they will often end up dictating which system you end up with based on their experience, specialty and financial incentives. The company from which you are buying the automation software knows the most about its systems so it is always helpful to have in-house assistance if they provide it. Certain automation systems provide custom programming and custom functions, so if your use case is specific or unique, a customizable platform will make a world of a difference in fully unlocking the potential of the document automation solution.

DOCUMENT AUTOMATION CHECKLIST 2020

COMPANY: _____

- | | | |
|---|--------------------------|--------------------------|
| 1. <u>Be web-based</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2. <u>Be a stand-alone app, not an add-on</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3. <u>Dynamically auto-generate interview questions</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4. <u>Modify multiple documents at once</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5. <u>Show the underlying document in real time</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 6. <u>Be able to make freestyle text edits</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 7. <u>Keep track of different versions and report differences</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 8. <u>Automate workflow of related tasks</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 9. <u>Be easy to maintain the templates</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 10. <u>Be able to create e-sign docs & closing binders</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 11. <u>Keep track of material terms of the contract</u> | <input type="checkbox"/> | <input type="checkbox"/> |
| 12. <u>Be easy to implement and to customize</u> | <input type="checkbox"/> | <input type="checkbox"/> |

Notes:
