Technische Universität München

Ingo Glaser, Tri Huynh, Oleksandra Klymenko,
Benedikt Labrenz, Prof. Dr. Florian Matthes

Legal Document Automation Tool Survey 2020

sebis

Technische Universität München

Ingo Glaser, Tri Huynh, Oleksandra Klymenko,
Benedikt Labrenz, Prof. Dr. Florian Matthes

Legal Document Automation Tool Survey 2020

Studie über Werkzeuge zur Automatisierung von Rechtsdokumenten 2020

Software Engineering for Business Information Systems (sebis)
Chair for Informatics 19
Technische Universität München
Boltzmannstraße 3, 85748 Garching b. München, Germany

About sebis

sebis is the chair for Software Engineering for Business Information Systems at the Institute for Informatics of the Technische Universität München. sebis has been established in 2002 with funding of the Ernst Denert-Stiftung and is headed by Professor Dr. Florian Matthes. The main research areas of sebis are:

- Strategic IT Management
- LegalTech (NLP, KR, ML, NLG)
- Service Platforms and Ecosystems
- Blockchain-Based Systems Engineering

Disclaimer

The information contained herein has been obtained from sources believed to be reliable at the time of publication but cannot be guaranteed. Please note that the findings, conclusions and recommendations that TU München, sebis delivers will be based on information gathered in good faith from both primary and secondary sources, whose accuracy cannot always be guaranteed by TU München, sebis.

TU München, sebis disclaims all warranties as to the accuracy, completeness, or adequacy of such information. TU München, sebis shall have no liability for errors, omissions, or inadequacies in the information contained herein or for interpretations thereof. In no event shall TU München, sebis be liable for any damage of any kind (e.g. direct, indirect, special, incidental, consequential, or punitive damages) whatsoever, including without limitation lost revenues, or lost profits, which may result from the use of these materials. The reader assumes sole responsibility for the selection of these materials to achieve its intended results.

There will be no warranty that the results of this publication will be economically and technically exploitable and unencumbered by third-party proprietary rights.

How to Cite

When referring to our study in publications please cite the following reference:

Glaser, I., Huynh T., Klymenko O., Labrenz B., Matthes, F.: Legal Document Automation Tool Survey 2020, *Technische Universität München*, Munich, Germany, 2020, available at: https://wwwmatthes.in.tum.de/pages/139spzlq343ft/Legal-Document-Automation-Tool-Survey.

r1

Abstract

Many legal documents are created manually in time and resource intensive processes that are not very efficient. Document automation aims to reduce this manual effort during the generation of documents by automatically assembling documents from a previously defined template. Legal documents are usually highly structured and standardized, opening up a large potential to apply document automation. This survey analyzes 13 major players in the market of document automation tools, namely ActiveDocs Opus, Berkeley Publisher, dox42, Lawlift, Legito, Logiforms, NintexWorkflow Cloud, Precisely, SmartDocuments, Templafy, Windward, Woodpecker and XpressDox to make them more comparable for practitioners and to provide tool vendors with an overview of requirements for a document automation tool in the legal domain.

Contents

Abstract vii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Structure of the Legal Document Automation Tool Survey . . . 1
- 1.3 Analyzed Document Automation Tools . . . 4

2 Criteria of the Legal Document Automation Tool Survey 7
- 2.1 Criteria for Core Document Automation Functionality . . . 7
- 2.2 Criteria for Supporting Features . . . 13

3 Summary 17

4 ActiveDocs Opus 25
- 4.1 Executive Summary . . . 25
- 4.2 Evaluation of Core Document Automation Functionality . . . 26
- 4.3 Evaluation of Supporting Features . . . 31

5 Berkeley Publisher 35
- 5.1 Executive Summary . . . 35
- 5.2 Evaluation of Core Document Automation Functionality . . . 36
- 5.3 Evaluation of Supporting Features . . . 41

6 dox42 43
- 6.1 Executive Summary . . . 43
- 6.2 Evaluation of Core Document Automation Functionality . . . 44
- 6.3 Evaluation of Supporting Features . . . 48

7 Lawlift 51
- 7.1 Executive Summary . . . 51
- 7.2 Evaluation of Core Document Automation Functionality . . . 51
- 7.3 Evaluation of Supporting Features . . . 56

8 Legito 59
- 8.1 Executive Summary . . . 59
- 8.2 Evaluation of Core Document Automation Functionality . . . 60

8.3 Evaluation of Supporting Features 65

9 Logiforms **67**
9.1 Executive Summary 67
9.2 Evaluation of Core Document Automation Functionality 68
9.3 Evaluation of Supporting Features 73

10 Nintex Workflow Cloud **75**
10.1 Executive Summary 75
10.2 Evaluation of Core Document Automation Functionality 77
10.3 Evaluation of Supporting Features 81

11 Precisely **83**
11.1 Executive Summary 83
11.2 Evaluation of Core Document Automation Functionality 84
11.3 Evaluation of Supporting Features 88

12 SmartDocuments **91**
12.1 Executive Summary 91
12.2 Evaluation of Core Document Automation Functionality 92
12.3 Evaluation of Supporting Features 97

13 Templafy **99**
13.1 Executive Summary 99
13.2 Evaluation of Core Document Automation Functionality 101
13.3 Evaluation of Supporting Features 104

14 Windward **107**
14.1 Executive Summary 107
14.2 Evaluation of Core Document Automation Functionality 108
14.3 Evaluation of Supporting Features 113

15 Woodpecker **115**
15.1 Executive Summary 115
15.2 Evaluation of Core Document Automation Functionality 116
15.3 Evaluation of Supporting Features 122

16 XpressDox **125**
16.1 Executive Summary 125
16.2 Evaluation of Core Document Automation Functionality 126
16.3 Evaluation of Supporting Features 130

17 Conclusion **133**

Bibliography **135**

1 Introduction

1.1 Motivation

Legal departments in companies and government agencies are creating large amounts of legal documents every day. In many cases this process is still heavily dependent on the manual labour of lawyers and support staff. Time and resources are lost on assembling each document by copying text blocks from existing documents and manually looking up the correct document structure.

The term *Document Automation* describes the trend of applying software solutions to automate the generation of documents [3, 2]. This study focuses at document generation as a sub-process of document automation. In most cases legal documents are highly structured and the decision which paragraphs are included depends on strict rules that have been defined in advance [7]. The underlying logic for the document structure is usually modified only when the law or regulations change. For this reason there is a high potential to use document automation in the legal domain [5, 1, 4].

The market for software tools for document automation consists of many vendors that offer tools which can automatically generate documents. Some of them solely focus on document generation while others integrate this functionality into a platform for automating business processes.

The goal of this *Legal Document Automation Tool Survey* is to evaluate and compare tools to give an overview of the market to practitioners who are interested in introducing document automation to their organization. Tool vendors are provided with requirements for document automation tools in the legal domain. For this reason, this study does not contain a ranking with clear winners and losers but provides a scorecard based on the requirements and corresponding scenarios that illustrate their practical applicability. The study serves as a source of reference for users to choose the appropriate document automation solution for their business needs.

1.2 Structure of the Legal Document Automation Tool Survey

The requirements, which were elicited based on real world use cases from the legal domain, are split into two categories to distinguish between core document automation func-

tionality and supporting features. Each requirement has a detailed description and is associated with a scenario that highlights for which organizations or use cases this requirement is especially important. Radar diagrams for each category of core requirements are used to visualize the results of the evaluation.

Core document functionality relates to features that are essential parts of the template creation process or document automation process. The category consists of the following requirements:

- Template Structure (see Section 2.1.1)
- Template Creation (see Section 2.1.2)
- Document Generation (see Section 2.1.3)
- Conditional Logic (see Section 2.1.4)
- Formulas (see Section 2.1.5)
- Documentation (see Section 2.1.6)
- Usability (see Section 2.1.7)

Supporting features are functionality that extend the core functionality of document automation tools and relate to the tool's integration into the existing landscape from a technical and an organizational point of view. The category consists of the following requirements:

- Integration (see Section 2.2.1)
- Document Management (see Section 2.2.2)
- User Management (see Section 2.2.3)
- Collaboration (see Section 2.2.4)
- Enterprise Scalability (see Section 2.2.5)
- Advanced Authentication (see Section 2.2.6)
- Data Ownership (see Section 2.2.7)

Name	Vendor	Headquarters	Evaluated
ActiveDocs Opus	ActiveDocs International Ltd.	Overland Park (USA)	Yes
Berkeley Bridge	Berkeley Bridge	Aalphen an der Rijn (NL)	Yes
Bryter	Bryter	Berlin (Germany)	No
Composer	Conga	Broomfield (USA)	No
Contract Express	Thomson Reuters	Toronto (Canada)	No
docmosis	Docmosis Pty Ltd.	Perth (Australia)	No
dox42	dox42 GmbH	Vienna (Austria)	Yes
ecrion	Ecrion Software	Rockville (USA)	No
GroupDocs.Assembly	Aspose Pty Ltd	Sydney (Australia)	No
HotDocs	AbacusNext	San Diego (USA)	No
Lawlift	LAWLIFT GmbH	Berlin (Germany)	Yes
Legito	Legito s.r.o.	New York (USA)	Yes
Litera Forte	Litera Microsystems	Chicago (USA)	No
Logiforms	Logiforms Software, Inc.	Vancouver (Canada)	Yes
Neonta	Neonta Logic	New York (USA)	No
Nintex Workflow Cloud	Nintex Global Ltd.	Bellevue (USA)	Yes
Precisely	Precisely AB	Gothenburg (Sweden)	Yes
SCHEMA ST4	SCHEMA Gruppe	Nuremberg (Germany)	No
SmartDocuments	SmartDocuments	Deventer (Netherlands)	Yes
Smartlaw	Wolters Kluwer	Cologne (Germany)	No
Templafy	Templafy ApS	Copenhagen (Denmark)	Yes
Vertragsgenerator	knowledgeTools	Berlin (Germany)	No
Windward	Windward Studios Inc.	Boulder (USA)	Yes
Woodpecker	Woodpecker Technologies, LLC	Boston (USA)	Yes
XpressDox	XpressDox (Pty) Ltd.	Cape Town (South Africa)	Yes

Table 1.1: List of identified tools

1.3 Analyzed Document Automation Tools

In July 2019 we contacted 25 vendors of document automation tools (see Table 1.1). The tools were identified via a global search, based on their online presence and secondary sources such as fellow researchers and practitioners. Out of the 25 contacted vendors, 13 were interested in participating in the study.

Table 1.2 lists the 13 evaluated tools, as well as their corresponding versions, vendors, and technical requirement in alphabetical order.

Tool	Version	Software category	Data ownership	Tech Req.
ActiveDocs Opus	Update 23	Web-based, MS Office Add-in	Cloud, On-premise	T
Berkeley Studio	4.8.9.3637	Web-based	On-premise	W
dox42	4.1.5.5	Web-based, MS Office Add-in	Cloud	T
Lawlift	Dated 18.10.2019	Web-based	Cloud	NT
Legito	Dated 09.09.2019	Web-based	Cloud	NT
Logiforms	3.1	Web-based	Cloud	NT
Nintex Workflow Cloud	Release 85	Web-based	Cloud	W
Precisely	Dated 19.09.2019	Web-based	Cloud	NT
SmartDocuments	2.19.12.01	Web-based, MS Office Add-in	Cloud	T
Templafy	5.6.0.1611	Web-based, MS Office Add-in	Cloud	NT
Windward Report Designer	16.5.0.97	MS Office Add-in	On-premise	T
Woodpecker	3.0.0.0	MS Office Add-in	Cloud	T
XpressDox	11.3.2.8	MS Office Add-in	Cloud	T

Table 1.2: List of evaluated tools. For Technical Requirement, the values are interpreted as: T - technical-user-oriented, NT - non-technical-user-oriented, and W - Workflow-based

2 Criteria of the Legal Document Automation Tool Survey

The evaluation criteria of the Legal Document Automation Survey are split into core document automation functionality and supporting features that are built around the document generation engine. Each core criterion is assigned a score ranging from 1 (low capability) to 5 (high capability). For the supporting document automation functionality, we only examine whether a considered advanced feature exists in the system or not, without assigning a score.

2.1 Criteria for Core Document Automation Functionality

This section describes the fundamental features for processing document automation tasks in legal context.

2.1.1 Template Structure

The criterion *Template Structure* evaluates how well document templates are structured and integrated with user inputs. The template structure should be unambiguous, easy to change and give as many options of formatting as possible while maintaining a previously defined style. It is important that dates can be displayed in a localized format inside the document.

Usage of user inputs inside the template should be as flexible as possible and centrally managed. Data types for common user inputs have to be supported, including:

- **Text field:** static text content of a template
- **Text input:** dynamic text spanned on a single line
- **Numeric text input:** similar to text input, but only numbers are accepted
- **Text area** (multiple line text input): dynamic text spanning across multiple lines
- **Date (time) picker:** a calendar (clock) interface for users to select the desired date (time)

- **Single selector**: an input interface which displays various options and allows user to choose only one as the answer
- **Multiple selector:** an input interface which displays various options and allows users to select multiple options as the answer
- **Table:** an interface to prompt data for creating a table, or convert structured or semi-structured data into table format

These inputs can be displayed as graphical components, tags, or functions in the graphical user interface (GUI).

Scenario: A big German insurance company manages its insurance policies with the help of a document automation tool. As insurance policies reflect a highly complex type of documents that contain varying legal clauses, the document automation tool shall support a vast amount of structural elements for templates. This must go beyond simple user inputs such as text input, date picker, or single selectors, but allow for reuse of sub-structures, insertion of tables and other highly dynamic and dependent elements.

2.1.2 Template Creation

The criterion *Template Creation* evaluates the process for creating a new template or updating an existing one. In particular, the possibility to conduct the following actions is analyzed:

- **Create template:** create a document containing static and dynamic text components. Content of dynamic texts is defined based on the user inputs
- **Edit template:** user shall be able to modify the structure and content of templates after their creation
- **Reuse template:** user shall be able to include the static and dynamic content from other templates in the current template
- **Export documents from PDF or DOCX files:** the tool shall be able to create a template from PDF or DOCX files containing (compatible) dynamic contents

Templates and parts of templates should be reusable including text blocks, template structures, style settings, and user inputs. There is a distinction between simply copying values and creating a reference to the original object that allows updating all objects at the same time. Templates should be managed centrally using a version control system and retain as much functionality for older versions as possible.

Scenario: The legal department of a large car manufacturer keeps a big amount of contracts with various suppliers. The contracts are created and stored by means of a document automation tool. Hereby, various contract templates share similar, and even the same clauses. Furthermore, some templates reuse structures of other templates partially. As the car manufacturer acts globally, the organization is subject to frequent changes in legislation and jurisdiction. This leads to frequent changes on text level, but also on a structural base. Hence, the tool shall support reuse of templates and respective parts and pieces of templates as well as editing functionality in a global manner.

2.1.3 Document Generation

The criterion *Document Generation* evaluates the process of manually generating a document from the user's perspective using a previously created template. The evaluation focuses on the number of steps to generate a document and on the important options for the file export.

Tools have to support exporting documents as a DOCX file and should allow exporting as a PDF file. Importing data from a database to substitute user inputs should be supported to reduce manual data input and allow bulk generation of multiple documents at the same time. Guidance of users through the document process is evaluated and can be based on annotations, warnings, and checklists. The evaluation also considers the user's control of the location of the generated document and if there are advanced functions like directly sending the document by mail.

Scenario: A big software vendor needs to create privacy policies for each offered tool. The data privacy officer (DPO) already collected all required information for each tool and stored them in a database. Furthermore, the DPO already created a dynamic template of a generic privacy policy. Now, each product owner generates the respective privacy policy by utilizing the provided template. In that process, the majority of tool-specific information shall be imported from the database into the template. The generated documents shall be reviewed in Microsoft Word before they will be sent out.

2.1.4 Conditional Logic

The criterion *Conditional Logic* evaluates how powerful conditional expressions can be and how flexible they can be used inside templates. A conditional expression in this context consists of a comparison of user inputs with static values or other user inputs. The conditional logic has to be able to model a dependence of one conditional expression on multiple user inputs by chaining comparisons with logical operators. Available operators for comparisons should depend on the data types of user inputs to prevent invalid expressions.

The value of a conditional expression should be able to decide whether one or multiple user input prompts or text blocks are displayed, to provide a value that can be used in

other parts of the template and to dynamically change the options of multiple choice user inputs. The evaluation includes how easy users can get an overview of the entire conditional logic used inside a template for validation.

The tools shall be able to process the comparison of the following data types:

- String
- Integer
- Decimal
- Boolean (true / false)
- Date
- Null (or equivalent such as checking a component is visible or is set)

Additionally, it is a considerable advantage if the tool can process arithmetic operations on dynamic numeric values. For example, in Germany, the amount of security deposit for property rental must not exceed the threefold amount of monthly rent. This can be illustrated in the following conditional expression:

$$deposit \leq monthly\_rent * 3$$

where *deposit* and *monthly_rent* are dynamic numeric variables user inputs.

In our study, we define four levels of complexity for conditional logic that document automation tools can handle. Level 1 is the most simple requirement, a tool can process a single conditional expression like $x \geq 12$. Level 2 requires a tool to handle a pair of chained conditional expressions, which comprises two conditional expressions and a conjunction (AND) or disjunction (OR) operator. For example:

$$x \geq 12 \; AND \; y \leq 20$$

To reach Level 3 complexity, a tool must solve a conditional statement comprising three chained conditional expressions, where two of them are nested. For instance:

$$(x \geq 12 \; AND \; y \leq 20) \; OR \; z = 27$$

Finally, at Level 4, a tool must be able to handle multiple chained conditional expressions with custom order-of-precedence, i.e. any group of conditional expressions can be nested. For example:

$$x \geq 12 \; AND \; ((y \leq 20 \; OR \; z == 27) \; AND \; k == \text{"Munich"}) \; AND \; (d \geq 52 \; OR \; d \leq 30)$$

Scenario: A law firm represents clients in civil law matters. The firm covers cases in various domains, but is specialized in compensation of pain. For each legal proceeding, the responsible lawyer needs to create an individual statement of claim for the underlying case. While the actual content of such a statement highly varies between cases, the overall structure shall be captured in a template. For that reason, a suitable document automation tool needs to be able to capture a vast variety of conditional logic (e.g.: the text needs to be adapted according to the victim's gender; depending on the age of the victim, the claim may change; each kind of accident requires different paths within the template; etc.)

2.1.5 Formulas

The criterion *Formulas* evaluates what kind of mathematical calculations can be included into templates. Formulas have to support basic arithmetic operators such as *add*, *subtract*, *divide*, and *multiply* and should support operators using strings and dates as input. The correct usage of operators should be unambiguous and users should be able to debug formulas. Additionally, advanced mathematical functions such as exponent, logarithm, ceiling, floor, and square root should be handled for extraordinary use cases.

Furthermore, the criterion Statistical Formulas evaluates the utilization of more advanced statistical operations such as min/max, average, median, mode, standard deviation or variance in order to handle very sophisticated use cases. Since basic arithmetic operations and advanced mathematical formulas are used more frequently in the legal domain, we separate the evaluation of these two criteria in our study.

Scenario: A medium-sized German software vendor manages its employment agreements by means of a document automation tool. The organization has different locations across Germany, as well as job positions with different skill levels required. As a result, each employment agreement usually constitutes a different salary based on the circumstances (kind of position, level of position, location of branch office). The document templates shall be capable of capturing such circumstances by utilizing proper advanced mathematical formulas in the document generation process.

2.1.6 Documentation

The criterion *Documentation* evaluates how good the available documentation supports non-technical users during their learning experience. The evaluation focuses on the completeness of the documentation, intuitive structuring and the availability of search tools. It is also considered whether examples and videos are available to the user as an illustration of how particular features work.

Scenario: A Bavarian judicial authority, which represents the Free State of Bavaria in all disputes before the courts of law, incorporates a document automation tool within their diverse processes. As an organization with limited access to technical personnel and budget for technical support or

training, the documentation of the document automation tool shall be tailored for non-technical users, but also be complete and comprehensive.

2.1.7 Usability

The criterion *Usability* evaluates the user experience from the perspective of non-technical users. It does not evaluate if functional tasks can be accomplished but how intuitive the process is for the user. This includes but is not limited to clean and concise user interfaces, intuitive designs of menus and user dialog windows, easy navigation between functionalities and not overloading screens with too much information. We conduct the usability evaluation based on the 10 usability heuristics for user interface design, proposed by Jakob Nielsen [6], namely:

1. **Visibility of system status**: the users are well informed about the current state of a system within a reasonable amount of time.
2. **Match between system and the real world**: the system uses familiar languages and conventions to communicate with users.
3. **User control and freedom**: undo and redo operations are possible as an ”emergency exit” from undesirable states.
4. **Consistency and standards**: any concept is described using the same languages and terms across different user interfaces of the system.
5. **Error prevention**: the system shall eliminate the possibility of user errors through validation and confirmation.
6. **Recognition rather than recall**: users should not remember how to use the system but naturally realize it.
7. **Flexibility and efficiency of use**: there are advanced interactions for expert users to accelerate execution processes in the system.
8. **Aesthetic and minimalist design**: only relevant and frequently used information is displayed.
9. **Help users recognize, diagnose, and recover from errors**: systems shall explain errors using concise, plain, and constructive messages to help users locate and fix them.
10. **Help and documentation**: availability of instructions on how to use features of the system.

This criterion reflects the subjective impressions that have been gathered during the evaluation.

Scenario: Susi, a lawyer with more than 40 years of professional experience in the domain of labour law has prepared and filed hundreds of lawsuits. Since Susi's law firm receives an increased number of client requests, they want to speed up their lawsuit preparation process by automating the generation of the statement of claim. For that reason, Susi needs to go through the whole process of a supporting document automation tool, starting with the template creation up to teaching her colleagues how to generate the actual statement by utilizing client data in the in-house database. As a non-technical user who is hesitant to adapt to a new tool, Susi values high usability in order to avoid rejection of the tool.

2.2 Criteria for Supporting Features

2.2.1 Integration

The criterion *Integration* evaluates the capabilities to trigger the document generation process programmatically by using an API or built-in integration. The evaluation focuses on the degree to which the functionalities of the tool can be accessed by other systems and how many constraints have to be respected.

Scenario: A large insurance company with thousands of customers manages a corresponding number of invoices. For invoicing a document automation tool shall be used. Different templates exist for the various kinds of invoices, which have been created manually. However, the invoices shall be created and sent out when due. As a result, the insurance policy management tool (IPMT) needs to automatically trigger the invoice creation based on its internal data. The IPMT shall also be able to gather the created document and send it out accordingly. Furthermore, if a bill is not paid, the IPMT creates an alert. This alert shall also be utilized to trigger another document generation process for the reminder of payment.

2.2.2 Document Management

The criterion *Document Management* evaluates to what degree document management functionalities are provided. These include archiving generated documents, tracking statuses or other attributes, version control, and organizing directories. A filterable list of all available documents should be provided with the option to make changes and create a new version.

Scenario: A medium-sized logistics company, Logit, transports many different goods within Europe. As a result, Logit has signed many contracts with its clients. As Logit is growing, it wants to utilize modern document automation software in order to create and manage the logistics contracts within the organization. As Logit currently has no document management tool in place, but uses a simple file system structure, the document automation tool needs to be able to provide a proper document management functionality.

2.2.3 User Management

The criterion *User Management* evaluates to what degree users can be managed centrally. This includes the creation of new users, management of user groups and central management of permissions to access templates and documents with different granularity.

Scenario: A big law firm with thousands of employed lawyers utilizes a document automation software in order to create contracts for clients. As the law firm deals with various clients from different domains, certain associates are only allowed to create contracts for specific purposes. For that reason, users should only be allowed to see and utilize a limited subset of templates and generated documents, depending on the user's role. Furthermore, specific master users need to be able to manage the user roles and access rights.

2.2.4 Collaboration

The criterion *Collaboration* evaluates in which ways teamwork is supported. This could be achieved by sharing documents with other users, tracking their changes or allowing communication via comments.

Scenario: A huge global software company wants to buy a smaller company, which created a valuable software. The global player hires a big law firm to perform the transaction. For that reason, a team consisting of 10 lawyers work on the transaction and negotiate the conditions with the small software company. During the negotiations, multiple versions of the contracts are drafted in collaboration. Each round of negotiations leads to a new draft of the contract. As different lawyers work on different aspects of the conditions, various changes in the contract may be required and be performed by different associates. Hence, the tool shall be able to properly support collaborative work.

2.2.5 Enterprise Scalability

The criterion *Enterprise Scalability* evaluates factors that are specific to enterprises. When dealing with a large number of users, templates and generated documents, it is important that tools scale organizationally and technically. Access to enterprise customer support that can not only deal with user problems but also with questions regarding integration or custom development is important.

Scenario: The legal department of a large car manufacturer wants to utilize a legal document automation tool. The tool needs to be integrated into the existing IT landscape. Furthermore, since the legal department owns not just a huge amount of legal documents, but is also subject to frequent changes with regard to legal relationships, it has high expectations for scalability and a continuous customer service.

2.2.6 Advanced Authentication

To help with the integration into the existing IT landscape and add extra protection to documents, additional authentication methods such as multi-factor authentication or single sign-on should be supported.

Scenario: A small law firm for IT law owns a diverse set of legal documents. This includes, but is not limited to client contracts, privacy policies, and terms of services. The law firm utilizes a document automation tool for the whole process of document generation, including template generation, up to the point where the data is stored and archived. As a result, the content of the tool is of high value for the law firm and used on a daily basis. As the law firm is not capable of a fundamental and secure IT landscape, advanced authentication is a must-have requirement. For the sake of ease of use, single sign-on should be supported. Furthermore, as users may access the tool from outside, such as at a client location, multi-factor authentication is also considered important.

2.2.7 Data Ownership

The criterion *Data Ownership* evaluates how much control users have over their templates and generated documents. In this study, we assess whether a tool is cloud-based or on-premise. A tool is considered as on-premise if it runs on local machines or on servers owned by the organization. For cloud-based tools the evaluation includes if templates can be downloaded and managed locally, the strength of isolation from other cloud tenants and if documents can be generated locally without uploading data to the cloud.

Scenario: A Bavarian judicial authority, which represents the Free State of Bavaria in all disputes before the courts of law, incorporates a document automation tool within their diverse processes. As the documents are highly confidential, legal constraints do not allow the authority to utilize external cloud providers for data storage purposes. As a result, the tool needs to be considered as on-premise, without any functionality to share documents with external parties. However, sharing templates with other authorities using the tool is highly demanded.

3 Summary

Table 3 provides an overview of the evaluation of criteria for core and supporting functionalities of the 13 tools included in this survey. The structure of the table is divided as follows: the assessment of *Template Structure, Template Creation, Document Management, Conditional Logic, Formulas, Documentation* and *Usability* criteria and their corresponding features; and evaluation criteria for supporting functionalities, namely *Integration, Document Management, User Management, Collaboration, Enterprise Scalability, Advanced Authentication and Data Ownership*. A detailed description for each criterion of any tool can be found in the corresponding sections in the tool evaluation chapter below.

For the assessment of core functionalities, a score band from 0 to 5 is assigned to each criterion. Features inside the same criterion are weighted equally because different user groups have various opinions regarding the importance of particular features depending on their business context. The weight is determined by taking 5 divided by the number of features. For example, the *Template Creation* criterion has five features, so each feature has a weight of 5 / 5 = 1, this means the maximum score each feature can obtain is 1.

On the other hand, the evaluation of supporting functionalities reflects the existence of considered advanced features in the system, so the result type is a Yes/No answer for each criterion or their features.

Criteria \ Tool	ActiveDocs Opus	Berkeley Studio	dox42	Lawlift	Legito	Logiforms	Nintex Workflow Cloud	Precisely	SmartDocuments	Templafy	Windward Report Designer	Woodpecker	XpressDox
Core Document Automation Features													
Template structure	**5**	**5**	**4.4**	**4.4**	**5**	**5**	**5**	**4.4**	**5**	**4.4**	**4.4**	**4.4**	**4.4**
Text label	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Text input	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Numeric input	✓	✓	✓	-	✓	✓	✓	✓	✓	-	✓	✓	✓
Text area	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Date (time) picker	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Single selector	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multiple selector	✓	✓	-	✓	✓	✓	✓	-	✓	✓	-	-	-
Table	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Template creation	**5**	**4.7**	**5**	**4.6**	**4.2**	**4.5**	**5**	**4.5**	**4**	**4.4**	**5**	**4.8**	**5**
Create basic template	1	1	1	1	1	1	1	1	1	1	1	1	1
Edit template	1	1	1	1	0.7	0.7	1	1	1	0.7	1	1	1
Reuse template	1	0.7	1	1	1	1	1	0.8	1	0.7	1	0.8	1
Create from PDF or DOCX	1	1	1	1	1	1	1	1	0	1	1	1	1
Template styling	1	1	1	0.6	0.5	0.8	1	0.7	1	1	1	1	1
Document generation	**5**	**5**	**5**	**2.5**	**5**	**2.5**	**5**	**5**	**5**	**5**	**5**	**5**	**5**
DOCX	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓
PDF	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓
Conditional logic	**5**	**5**	**5**	**5**	**5**	**5**	**5**	**1.3**	**2.5**	**1.3**	**5**	**2.5**	**5**
Level 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Level 2	✓	✓	✓	✓	✓	✓	✓	-	✓	-	✓	✓	✓
Level 3	✓	✓	✓	✓	✓	✓	✓	-	-	-	✓	-	✓
Level 4	✓	✓	✓	✓	✓	✓	✓	-	-	-	✓	-	✓
Mathematical formulas	**5**	**3.3**	**3.3**	**0**	**5**	**2.5**	**1.7**	**0**	**0.8**	**0**	**4.2**	**5**	**3.3**
Basic (+ - x)	✓	✓	✓	-	✓	✓	✓	-	✓	-	✓	✓	✓
Advanced													
Square root	✓	✓	✓	-	✓	✓	-	-	-	-	✓	✓	-
Logarithm	✓	-	✓	-	✓	-	-	-	-	-	-	✓	-

Criteria \ Tool	ActiveDocs Opus	Berkeley Studio	dox42	Lawlift	Legito	Logiforms	Nintex Workflow Cloud	Precisely	SmartDocuments	Templafy	Windward Report Designer	Woodpecker	XpressDox
Exponent	✓	✓	✓	-	✓	✓	✓	-	-	-	✓	✓	✓
Ceiling	✓	✓	-	-	✓	-	-	-	-	-	✓	✓	✓
Floor	✓	-	-	-	✓	-	-	-	-	-	✓	✓	✓
Statistical functions	**0.8**	**0.8**	**1.7**	**0**	**5**	**5**	**1.7**	**0**	**0**	**0**	**3.3**	**5**	**0.8**
Min/Max	✓	✓	✓	-	✓	✓	✓	-	-	-	✓	✓	✓
Average	-	-	✓	-	✓	✓	✓	-	-	-	✓	✓	-
Median	-	-	-	-	✓	✓	-	-	-	-	✓	✓	-
Mode	-	-	-	-	✓	✓	-	-	-	-	-	✓	-
Standard deviation	-	-	-	-	✓	✓	-	-	-	-	✓	✓	-
Variance	-	-	-	-	✓	✓	-	-	-	-	✓	✓	-
Documentation	**5**	**4**	**3**	**4.9**	**3.6**	**5**	**4.6**	**3.9**	**3.6**	**4.7**	**4.7**	**5**	**5**
Content	1	0.5	0.7	1	0.8	1	1	1	0.8	1	1	1	1
Search	1	0.9	0	0.9	0.8	1	1	0.9	0.8	1	0.7	1	1
Navigation	1	1	0.8	1	0	1	1	1	1	1	1	1	1
Instruction videos	1	0.6	0.5	1	1	1	0.6	0	0	1	1	1	1
Examples	1	1	1	1	1	1	1	1	1	1	1	1	1
Usability	**4.8**	**4.3**	**3.8**	**3.6**	**3.7**	**4**	**4.2**	**4.9**	**2.4**	**4.2**	**3.4**	**4.5**	**3.8**
Visibility of system status	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.3	0.5	0.5	0.5	0.5
Match between system and real world	0.5	0.3	0.4	0.2	0.4	0.4	0.3	0.5	0.3	0.5	0.1	0.4	0.3
User control and freedom	0.5	0.4	0.4	0.4	0.2	0.2	0.4	0.5	0.1	0.5	0.4	0.4	0.3
Consistency and standards	0.5	0.4	0.5	0.5	0.5	0.5	0.5	0.5	0.4	0.5	0.4	0.5	0.5
Error prevention	0.5	0.5	0.3	0.2	0.4	0.5	0.5	0.5	0.5	0.3	0.4	0.5	0.3
Recognition rather than recall	0.4	0.4	0.3	0.4	0.5	0.3	0.3	0.5	0.3	0.4	0.2	0.5	0.3
Flexibility and efficiency of use	0.5	0.5	0.3	0.2	0.2	0.4	0.2	0.5	0.2	0.5	0.3	0.5	0.4

Criteria \ Tool	ActiveDocs Opus	Berkeley Studio	dox42	Lawlift	Legito	Logiforms	Nintex Workflow Cloud	Precisely	SmartDocuments	Templafy	Windward Report Designer	Woodpecker	XpressDox
Aesthetic and minimalist design	0.4	0.5	0.5	0.4	0.5	0.5	0.5	0.5	0.2	0.5	0.3	0.5	0.3
Recognize, diagnose and recover from errors	0.5	0.4	0.3	0.3	0.2	0.2	0.5	0.5	0.1	0.2	0.3	0.3	0.4
Help and Documentation	0.5	0.4	0.3	0.5	0.3	0.5	0.5	0.4	0	0.3	0.5	0.4	0.5
Supporting Features													
Integration	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Version control	✓	✓	-	✓	✓	-	-	-	✓	-	-	✓	-
User management													
User group management	✓	-	✓	✓	✓	✓	-	✓	✓	✓	-	✓	✓
Access right management	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
Collaboration	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Authentication													
Two-Factor Authentication													
OTP via SMS	-	-	-	-	-	-	-	-	-	-	-	-	-
OTP via email	-	-	-	✓	-	-	-	-	-	✓	-	-	-
via mobile app	-	-	✓	-	-	-	-	✓	-	✓	✓	✓	✓
Single Sign-On	✓	-	✓	-	-	-	✓	-	✓	✓	✓	✓	✓
Data ownership													
Cloud-based	✓	-	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓
On-premise	✓	✓	✓	-	✓	-	-	-	✓	-	✓	-	-

Figure 3.1: Evaluation of eight Core Document Automation Functionalities for 13 tools, each tool has its own bubble color. Bubble size indicates the score of a tool in a specific evaluation criterion.

From the evaluation, all the tools display a remarkable performance in the *Template Creation* and *Template Structure* criteria. Specifically, most of the tools support all the input types considered in this study. In addition, all of the tools excel at *Template Creation* by providing features for creating, editing, reusing, and importing PDF or DOCX files as new templates. Regarding the file formats of generated documents, except for Lawlift and Logiforms which can generate only DOCX and PDF documents respectively, all other tools can generate both PDF and DOCX files from templates.

Support for conditional expressions and formulas varies significantly among document automation tools. Logiforms, Legito, Lawlift, ActiveDocs, Nintex Workflow, dox42, Windward, Berkeley Publisher and XpressDox can handle the most complex nested and chained conditional expressions. Note that although Lawlift can handle complicated conditional statements, users can only apply logical expressions based on answers from multiple choice question type. Woodpecker can handle chained conditional statements by default, however modeling compound logical expressions requires additional effort from users to define multiple placeholders to group conditional statements and then join these placeholders with AND/OR operators. Precisely and SmartDocuments provide limited support to this criterion by handling one logical expression per conditional statement.

In mathematical formula evaluation, Woodpecker excels at handling basic arithmetic operations, as well as advanced and statistical mathematical functions considered in this study. Although Legito does not support all statistical functions, the tool can handle basic arithmetic and all advanced mathematical formulas. Precisely, Lawlift, dox42, and SmartDocuments provide no support for constructing mathematical expressions. Other tools can handle several advanced formulas considered in this study along with addition, subtraction, multiplication, and division operators.

Documentation is well prepared by the majority of the tools with structured, searchable and detailed description of document automation features along with explanatory videos and examples. Several tools need certain improvement in their documentation content. Specifically, vendors can structure their documentation and improve searchability to enhance user experience in searching features. Importantly, documentation shall be constantly updated to avoid giving outdated instructions to users.

In terms of usability, about half of the surveyed tools can be considered user-friendly with regards to the 10 Nielsen's heuristics. To a certain degree, these tools demonstrate highly responsive, intuitive, consistent, error preventive, flexible and minimalist user interface designs along with informative documentation and error messages. For other tools, improvement areas include, but are not limited to 1) validating inputs to prevent erroneous data from entering the system; 2) providing undo and redo functions for users to exit un-

desirable system states; 3) announcing diagnostic error messages with simple language to help users fix problems; 4) reducing cognitive load for users with straightforward and illustrative user interfaces; and 5) redesigning or providing optional workflows which are intuitive to non-technical users.

Regarding supporting features, for integration to external systems, ActiveDocs, dox42, Lawlift, Logiforms, Nintex Workflow, Precisely, Templafy, Windward, Woodpecker and XpressDox provide APIs for servers or software to use their document automation features. 6 out of 13 tools support version control for users to keep track of previous editions of their templates, hence users can revert their documents to preceding versions when necessary. This feature is available in Berkeley Publisher, Lawlift, Legito, Nintex Workflow, SmartDocument, and Woodpecker.

User Management and Collaboration features are supported by most of the tools. Templafy, SmartDocuments, ActiveDocs, Lawlift, Legito, Logiforms and Nintex Workflow and Precisely provide their own user interfaces for collaboration, managing user groups and defining read or write access to authorize users. On the other hand, other MS Word Add-In tools leverage the user management, document sharing, comment, and chat functionalities developed by Microsoft. The tools which belong to this group are dox42, Windward, Woodpecker, and XpressDox.

Two-factor authentication is considered to provide extra protection to documents. Precisely and Templafy offer this feature via mobile app authenticators. Templafy also provides extra authentication via email. As MS Word Add-Ins, XpressDox, Woodpecker, Windward and dox42 inherit Two-Factor Authentication functionality developed by Microsoft. Users need to download the Microsoft Authenticator app to their mobile phone to access this service.

For user groups that prefer on-premise tools, Berkeley Publisher and Windward are the appropriate options as they do not use cloud facilities for automatically generating documents. Other tools entirely or partly depend on their cloud platforms to execute document automation features.

From the technical requirements perspective, the 13 tools can be classified into 3 groups:

1. **Non-technical-user-oriented**: tools in this category do not require users to have prior technical knowledge of software development, database, or spreadsheet for generating documents. Users can quickly and intuitively create templates, configure where dynamic data are displayed in templates, and command the tool to generate corresponding documents. These tools are Logiforms, Legito, Lawlift, Woodpecker, Precisely, and Templafy.
2. **Technical-user-oriented**: These tools require certain technical experience to use doc-

ument automation features such as configuring dynamic elements in templates, inputting data sources, defining conditional logic and mathematical formulas using programming languages. Advanced users can quickly learn or leverage features of tools in this classification, yet for non-technical users it can be considerably challenging. ActiveDocs, dox42, SmartDocuments, Windward, and XpressDox belong to this category.

3. **Workflow-based**: Users define workflows to configure their document automation process, declare template location, create a questionnaire, update dynamic elements in templates based on answers from the questionnaire, and generate documents. Although modelling process can be an intuitive visualization for non-technical users, the actual configuration of workflow content requires certain technical knowledge, so it can be difficult at first. This issue can be addressed with the help of documentation and frequent usage of the features. Nintex Workflow and Berkeley Publisher belong to this group.

For straightforward, intuitive document generation, one can consider using **Non-technical-user-oriented** tools. In contrast, **Technical-user-oriented** tools can handle use cases which demand data sources, sophisticated logical expressions, advanced technical users and API integration to external systems. For users who would like to plan their document automation procedure as process models, **Workflow-based** tools provide customizable and manageable workflows for users to configure the document generation process based on business requirements. It is recommended that users read the documentation of these tools first to obtain a complete overview of their user interface and logic.

4 ActiveDocs Opus

ActiveDocs Opus (in the following called ActiveDocs) is a document automation tool that can be run both in the cloud and on premises, giving users full control over the system and data. The basic setup consists of a Microsoft Word Add-In for creating templates and a content manager for managing internal resources that are installed on the user machine. ActiveDocs is developed by the company with the same name since 2002.

4.1 Executive Summary

ActiveDocs is a powerful tool with a high degree of customization and reusability of templates. This characteristic is reflected on an all-inclusive user interface, which can be complex to new users, especially for non-technical ones. The tool supports definitions of advanced logical expressions and mathematical formulas during the template construction process. All input types required in this study can be declared, validated and masked in ActiveDocs. Besides, users can query and filter results from data sources to customize their output.

The process of automating documents starts with defining dynamic input elements. This is done by creating questionnaires or extracting information from data sources, e.g. databases. Next, users place dynamic input elements in specific positions of their templates. Finally, end-users fill in the questionnaires or trigger the document automation process to obtain a PDF or DOCX file.

ActiveDocs provides access rights and user management features to templates and generated documents. The tool also provides APIs for external systems to use its document automation features.

Overall, the tool delivers flexible, multi-functional document automation features which are capable of handling complicated template structures. However, frequent usage may be necessary for new users to familiarize themselves with the features included in the tool's user interface.

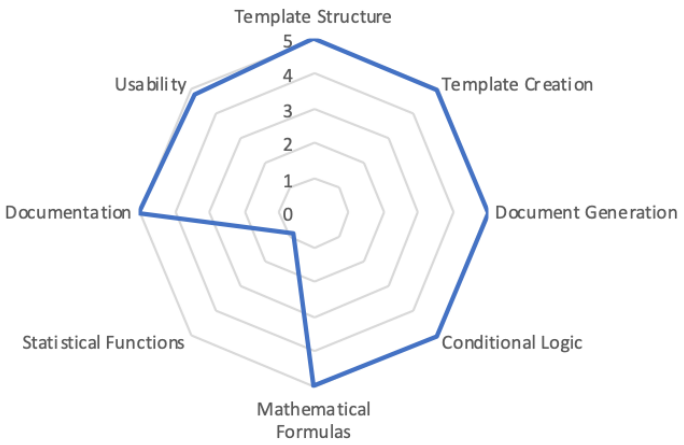

Figure 4.1: Radar diagram for ActiveDocs - Core Document Automation Functionality

Integration	✓
Version Control	✓
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	-
○ Single Sign-On	✓
Data Ownership	
○ Cloud-based	✓
○ On-premise	✓

Table 4.1: Supporting features of ActiveDocs

4.2 Evaluation of Core Document Automation Functionality

A template in ActiveDocs consists of document(s) containing dynamic fields and a questionnaire. A questionnaire consists of any number of questions split among multiple pages. Each question is uniquely identified by a name and can contain a default value. Available data types for questions are listed in table 4.2.

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 4.2: Template structure features of ActiveDocs

Every question regardless of data type can allow free user input or can only accept a set of predefined options. Predefined options can be displayed as a dropdown menu or a set of radio buttons where only one option can be selected at the same time, or a set of checkboxes that allows for multiple options to be selected at the same time. The available options can be static or be fetched from a data source like a database or a file.

The document body of a template consists of static text containing fields whose value depends on user inputs for questions. Formatting like paragraphs, page breaks, and clause lists can be applied to the template.

Templates in ActiveDocs can model any user inputs in the scenario and use them in a flexible way inside the document.

Score: 5/5

4.2.1 Template Creation

Templates are created with a Microsoft Word Add-In. Templates and reusable template parts are managed in a standalone tool running on the user‘s local machine. A new template can be created from scratch or imported from dotm files (a macro-enabled docx file) or existing templates.

Users can write static text and add dynamic input fields into templates. Text blocks and single words can be made conditional (see Section 4.2.3) and every value of a field or question can be manipulated using formulas (see Section 4.2.4).

Text blocks that may contain fields or conditional statements can be saved in a centrally managed repository and reused in other templates. Similarly, sets of questions can also

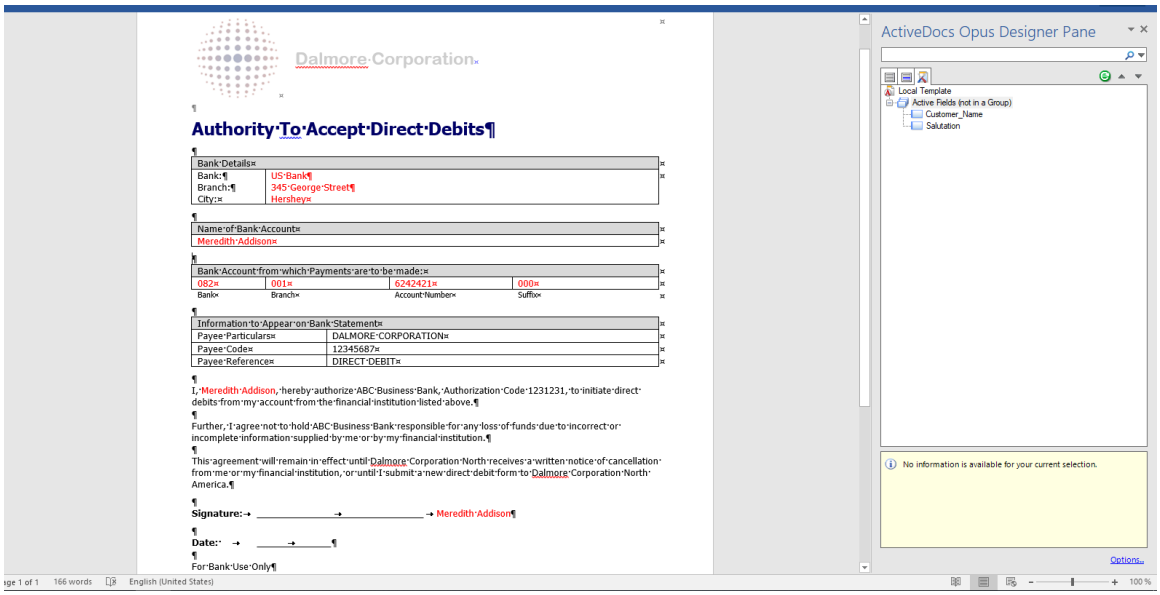

Figure 4.2: ActiveDocs - Template Structure

be saved and reused in other questionnaires. Changes that are made to reusable template parts are automatically propagated to all templates in which they are used.

After a template or reusable template part has been changed, it has to be checked in before the changes take effect. This creates a new version in ActiveDocs' version management while old versions can still be accessed and used. Most components of templates in ActiveDocs are centrally managed and can be reused, saving a lot of time in creating and updating templates.

Score: 5/5

4.2.2 Document Generation

The document generation process in ActiveDocs can be customized to a high degree. In a basic scenario, users answer all questions from the template which are split over multiple pages and can be annotated with explaining help texts. User inputs can be partially or completely substituted by external data sources like databases, files or API calls to any REST or SOAP API.

After submitting the questionnaire, the answers are stored in a database and users can choose between a set of delivery options for the generated document depending on the template's customization. This may include sending the document as an attachment of an

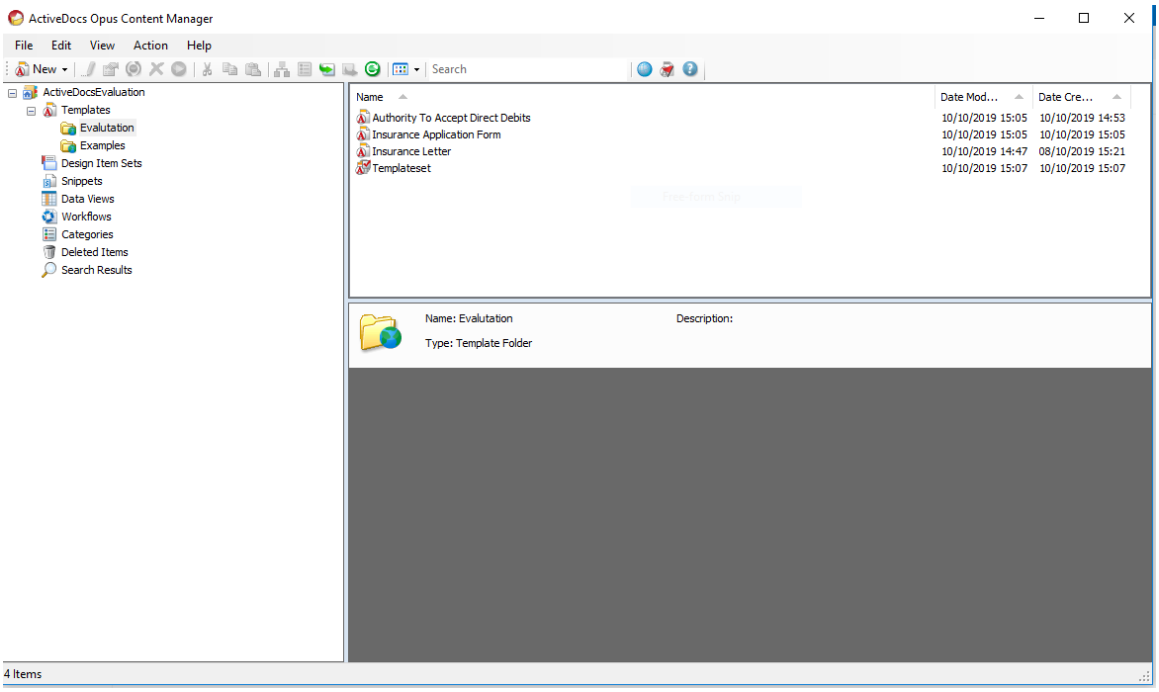

Figure 4.3: ActiveDocs - Component Management

email, triggering an E-Signing process or downloading the document. A variety of file formats are supported including DOCX and PDF. Depending on the use case, more complex workflows can also be triggered. For instance, multiple templates can be generated as a group or merged into a single file.

Documents can be generated using a combination of data sources including manual user inputs resulting in a seamless document automation process. ActiveDocs can import data sources from MS Excel, MS Access, SQL Server, OLE-DB or ODBC-compliant databases, and cloud applications.

Score: 5/5

4.2.3 Conditional Logic

Conditional logic can be used to decide if certain text blocks or words should be displayed. Conditions are reusable within a template and can be nested with the inner condition being only evaluated if the outer condition is true.

A simple conditional expression consists of a comparison of any user input value with another user input or a static value. Conditional expressions can be chained using ”AND”

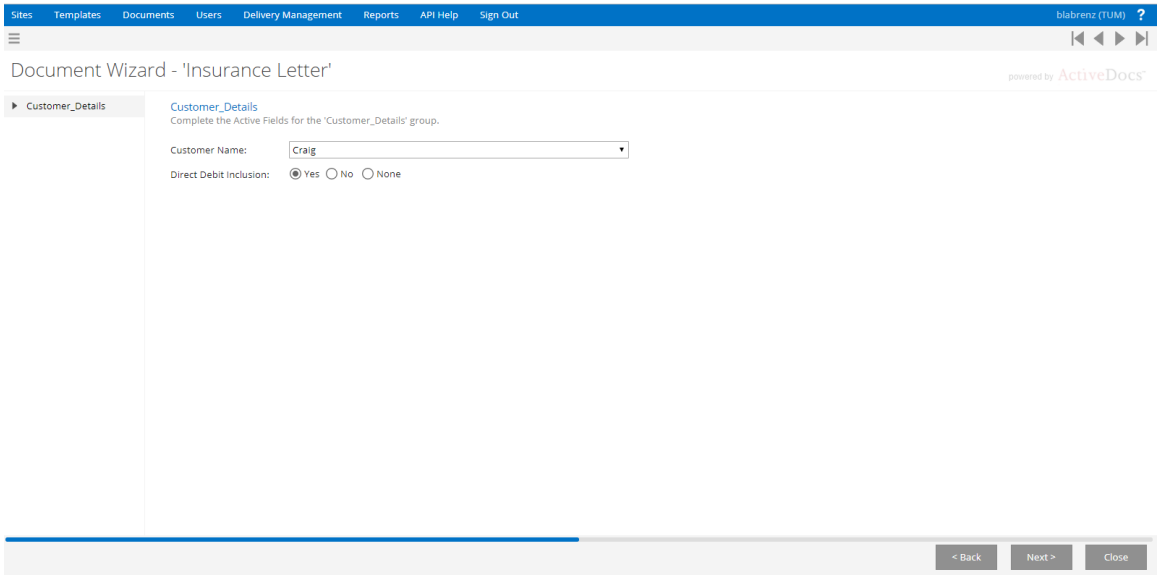

Figure 4.4: ActiveDocs - Document Generation

or ”OR” operators but not a combination of both. Advanced conditional expressions utilize a free text formula containing any number of nested comparisons or other expressions (see Section 4.2.4). Additionally, formulas can be used to model conditional expressions by utilizing if-then-else syntax.

Correct behavior of complex conditional expressions can be hard to validate. In ActiveDocs, the validation is made easier by a testing feature that checks for syntax errors and shows the result of logical expressions from supplied test inputs.

ActiveDocs supports advanced logical expressions to model and reuse complex decision logic with a user-friendly interface for both advanced and non-technical users.

Score: 5/5

4.2.4 Formulas

Values in fields and conditional expressions can be changed using a mathematical calculation. Formulas in ActiveDocs are based on the VBScript language with a restricted syntax. Operators include but are not limited to arithmetic and conditional operators. Similarly to complex conditional expressions they can be validated using a testing feature.

Formulas in ActiveDocs are flexible and powerful to apply and validate both basic and advanced mathematical operations.

Basic and Advanced Formulas : **5/5**
Statistical Functions : **0.8/5**

4.2.5 Documentation

The documentation consists of an online e-learning center with tutorial videos, and an integrated training manual with help pages. The descriptions aims at non-technical users and are very detailed. Documentation is well-structured by feature groups and is searchable.

Score: 5/5

4.2.6 Usability

ActiveDocs offers various options to insert elements into templates. The user interface is aesthetic, but due to extensive functionalities, several graphical elements have complex designs which may not be intuitive to users at first sight. ActiveDocs also provides advanced error prevention mechanisms to help users avoid syntax errors. In terms of consistency and standards, ActiveDocs' user interface is similar to Windows applications, which makes the process of working with the tool more intuitive for the users. In conclusion, ActiveDocs offers flexible and powerful features for document automation, but it requires certain effort from users to familiarize themselves with the system.

Score 4.8/5

4.3 Evaluation of Supporting Features

4.3.1 Integration

ActiveDocs provides a RESTful API with endpoints for all important functionalities like managing templates, generating documents from templates, accessing documents in the document management and running a diagnostics check on the server.

4.3.2 Document Management

After a document has been generated from a template, it is saved in an internal database. A list of all documents that a user has access to is available and can be filtered by the organization that owns it. Documents can be downloaded, updated, deleted and recreated with the latest version of the template. Previous versions of a document can be accessed and downloaded. More options like sending the document as an attachment of an email

are available through configurations.

ActiveDocs offers many useful functionalities for managing documents and only falls short in limited capabilities to filter large numbers of documents.

4.3.3 User Management

Administrators can create and manage users and user groups. Permissions for users and user groups can be set separately for each feature and many options. Similarly, the user interface can be customized for each user and user group to only display buttons and options the user should see. Management of users and user groups can be integrated with Microsoft Active Directory importing the existing accounts and settings.

Permissions can be set on a maximum granularity which allows users to map all organizational structures to user roles.

4.3.4 Collaboration

Generated documents can be viewed by all users in an internal organization while being assigned to a single user. They can be reassigned to another user who can create new versions of the documents. Documents can also be reassigned during the document generation process allowing teams to answer questions in collaboration. An approval process can be applied to new and updated templates or template parts to let supervisors validate them before they can be used in a productive environment. Communication between users within the tool is not possible.

4.3.5 Enterprise Scalability

ActiveDocs offers options of customization for most features to fit a complex business logic. Technical support is available to help with user problems, setup of an ActiveDocs server and regular maintenance.

4.3.6 Advanced Authentication

The tool offers Single Sign-On and user management through Microsoft Active Directory to provide a seamless integration with other services in enterprises.

4.3.7 Data Ownership

ActiveDocs gives users full control over the system and data, providing flexible deployment options. The tool can be run both in cloud and on-premise. Most often, ActiveDocs is deployed in public or private cloud, however, users can also deploy it on-premise by installing the software in virtualized environments. The vendor also offers the possibility to

install ActiveDocs on physical servers without virtualization, however, this option is less common as it has proven to be less efficient in comparison to other deployment options.

5 Berkeley Publisher

Berkeley Publisher (in the following called Berkeley) is a document automation and decision support tool that is hosted in the cloud or installed on-premises. It is complemented by a standalone editor that is used to create templates. The tool is developed by Berkeley Bridge since 2015.

5.1 Executive Summary

Users generate documents in Berkeley Publisher by defining workflows. The tool supports all the input types considered in this study, along with declarations of complex logical expressions and certain advanced mathematical formulas.

The document automation process starts with creating questionnaires of dynamic input elements. Users need to create a text fragment for each dynamic input, and refer these fragments to their Word templates. Afterwards, users save and publish their workflows as a decision tree model. At this point, running the model requires users to fill in the questionnaires. Finally, the tool generates a document based on the defined workflow, templates and answers from the questionnaires.

In terms of usability, configuring workflows in decision trees is not intuitive for non-technical users. Nevertheless, the process of creating questionnaires is straightforward. Additionally, Berkeley Publisher develops a minimalist interface that is user-friendly to both advanced and non-technical users. The tool also offers error prevention mechanisms to filter invalid inputs from entering the document automation process. Noticeably, documentation is self-explanatory, but several chapters are outdated.

To summarize, Berkeley Publisher is an on-premise tool which utilizes workflows to generate documents. The tool requires certain effort from users to familiarize themselves with the system. Berkeley Publisher also has a version control system to help users track or revert back to previous models.

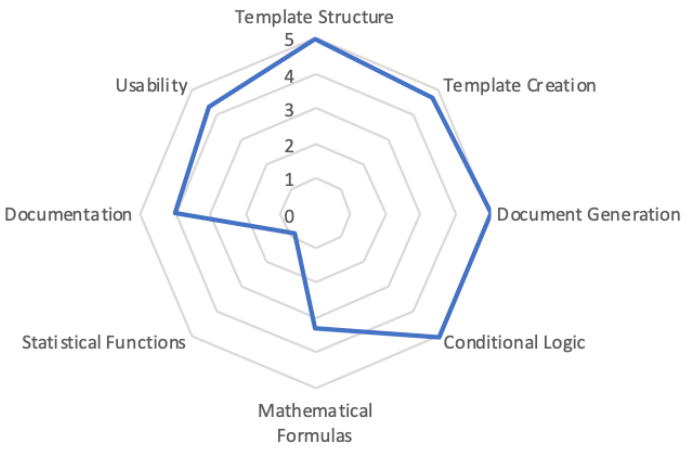

Figure 5.1: Radar diagram for Berkeley - Core Document Automation Functionality

Integration	-
Version Control	✓
User Management	
○ User Group Management	-
○ Access Rights Management	-
Collaboration	-
Advanced Authentication	-
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	-
○ Single Sign-On	-
Data Ownership	
○ Cloud-based	-
○ On-premise	✓

Table 5.1: Supporting features of Berkeley Publisher

5.2 Evaluation of Core Document Automation Functionality

5.2.1 Template Structure

Templates in Berkeley Publisher are made up of a decision tree and one or more document files that are attached to it. Nodes in the decision tree can contain questions, text blocks calculations (see Section 5.2.5), explanatory text and an action that generates the document.

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 5.2: Template structure features of Berkeley Publisher

Questions can have a default value and can be marked as required or read-only. There are different types of questions available as listed in table 5.2.

The document file is a DOCX file containing static text annotated with XML tags that are replaced by text fragments during the document generation. Text fragments are defined in nodes of the decision tree, they contain static text and/or values from answers to questions. Users can also enforce conditional statements on text fragments.

The decision trees in Berkeley are very flexible, which provides various approaches to model the same template.

Score: 5/5

5.2.2 Template Creation

To create a new template, one node in a decision tree must contain at least one existing DOCX file that will serve as the document body of a template. Decision trees are constructed by creating new nodes and linking them together. Questions, definitions of text blocks and formulas can be attached to nodes. Links between nodes, text blocks and formulas can be made conditional by attaching a conditional rule to them (see Section 5.2.4). XML tags can be generated for each text block and have to be manually inserted into the document file at the desired position using Microsoft Word.

A filterable list of all text blocks that are used in the template is available at all times and can be exported with all attached conditional rules as a CSV file for external validation. Whole decision trees or a subtree can be saved to a repository on the server and inserted into other templates. If a template structure is updated, changes are automatically propagated to all templates in which it is used. After a template is created or updated, it can be published to a server creating a new version in the version management.

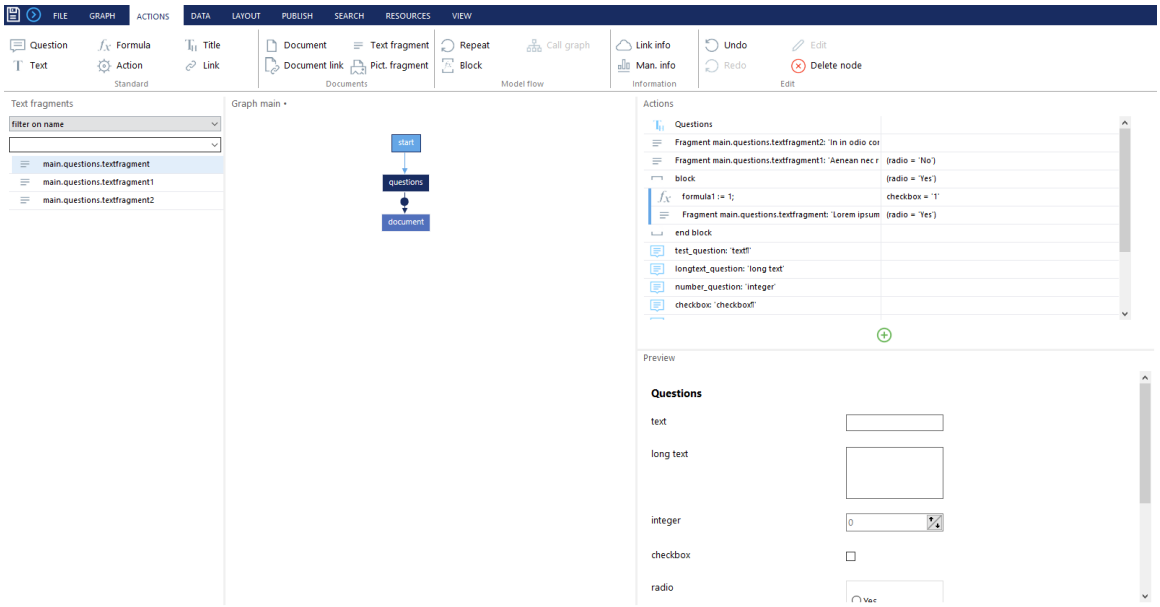

Figure 5.2: Berkeley - Template Structure

While the process of creating a decision tree is optimized and works smoothly, inserting text blocks as XML tags into the document body is a manual task, and not intuitive to non-technical users. Additionally, at the time of this survey, inserting text blocks into Word by drag-and-drop does not work. Users need to change the element name of the tag to *textfragment* so that the answers of questions can be linked properly to the template.

Score: 4.7/5

5.2.3 Document Generation

The manual document generation process can be triggered from the server and the editor in which templates are created. The process begins at the start node and progresses through the decision tree according to links. A visualization of the decision tree and the current position in it can be displayed at all times. If the current node contains any question, users are presented with a questionnaire and the process continues after the questionnaire is submitted.

After the last questionnaire is submitted, one or more documents are generated and can be downloaded as DOCX or PDF file based on the template configuration. User inputs can be complemented by calls to databases if the configuration of the template allows it.

The document generation process in Berkeley depends on the template's decision tree and offers a large degree of customization.

Score: 5/5

5.2.4 Conditional Logic

Conditional expressions can be used to determine if text blocks are included in the document, if formulas are applied and if nodes in the decision tree are visited. A conditional expression contains any number of comparisons between a value from a question or calculation and a static value or another answer or calculation. Supported operators for comparisons are =, >, <, >=, <= and <>. Comparisons can be nested using brackets and the logical operators "and" or "or". The syntax of conditional expressions is automatically validated, expressions with syntax errors cannot be saved but the error is not highlighted, creating difficulties for users to fix it.

A less powerful conditional logic is available in document files. XML tags can be inserted to determine if a text block, which can contain more XML tags, will be included in the generated document. Only a comparison between a single value from a question or formula and a static value can be used in this type of conditional expression.

Conditional logic in Berkeley can be attached to every relevant component of a template and can model complex business logic.

Score: 5/5

5.2.5 Formulas

Formulas can be added to nodes or used inside of conditional expressions. Available functions are shown in a structured list and include basic arithmetic functions, square root, ceiling, exponent, string and date manipulation. A short explanation of the correct usage and expected input parameters are displayed during the formula creation. The syntax of formulas is automatically validated and formulas with syntax errors cannot be saved but the mistake is not highlighted.

Formulas in Berkeley Publisher are easy to use and provide a wide range of available functions. Nevertheless, error messages do not reveal the locations and reasons of mistakes, so diagnosing and fixing syntax errors are not straightforward.

Basic and Advanced Formulas	:	**3.3/5**
Statistical Functions	:	**0.8/5**

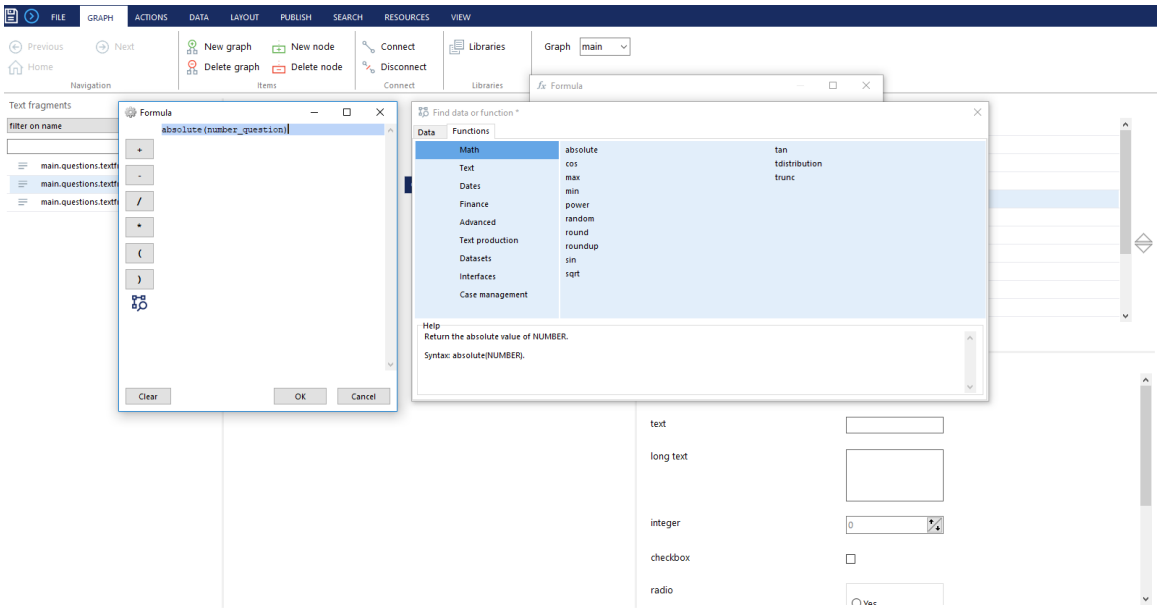

Figure 5.3: Berkeley - Formulas

5.2.6 Documentation

The documentation for Berkeley consists of a well-structured and searchable list of articles describing all features in detail along with videos and examples. It is written mostly for non-technical users with a section for developers. Noticeably, the description of how to add references to answers of questions in Word is outdated at the time this survey is conducted.

Score: 4/5

5.2.7 Usability

The user interface is minimalist and reduced to a minimum level of complexity. Advanced features are hidden by default but can be found where users would expect them. Building decision trees and adding actions to them is not very intuitive for non-technical users. However, if the template is configured correctly during the workflow construction phase, generating documents from templates works as expected. Dynamic elements can be inserted into Word Documents by using drag-and-drop mechanism or by manually typing XML text fragment tags. However, drag-and-drop mechanism does not work as expected for text fragments, i.e. when dragging the text fragment into Word, Berkeley Publisher interprets it as an XML tag of text input, so the content of the fragment is not displayed in the template. To fix this, users need to manually change the name of the dragged XML

tags from text input to text fragment. Another advantage of the tool is validation of user form inputs, numeric and date input fields, which do not accept invalid characters, thus typing mistakes are considerably prevented.

On the other hand, Berkeley Publisher gives hints to users about the expected data type or name of the referenced input element while writing logical expressions, but it does not display meaningful error messages when defining logical expressions. Although the tool refuses invalid logical expressions, there is no message explaining the type or position of the error. This may discomfort users when they make mistakes while writing complex logical conditions in the system.

Score 4.3/5

5.3 Evaluation of Supporting Features

5.3.1 Integration

The Berkeley Publisher server provides a SOAP API to trigger the document generation process.

5.3.2 Document Management

Berkeley Publisher does not offer a document management system.

5.3.3 User Management

Administrators can create, edit and delete users and user groups on the Berkeley server. There are four different types of user roles:

- administrator: Can access all features except for the data generated by templates.
- user: Can publish and administer own templates.
- guest: Can access templates.
- viewer: Can use data generated by templates for analytics.

User groups are used to determine which templates users have access to. The template editor currently does not support user management.

User roles in Berkeley Publisher can be used to model most organizational structures.

5.3.4 Collaboration

Decision trees and subtrees in the central repository each have an owner that is responsible for keeping it up-to-date. This can help splitting up work in large teams. During the template creation and document generation capabilities for collaboration are limited because communication and sharing of files that have not yet been published has to happen outside of the tool.

5.3.5 Enterprise Scalability

Decision trees in Berkeley Publisher can model any complexities of large organizations and can be modularized to enable centralized management of reusable resources. The tool can be easily integrated into the existing IT landscape with an on-premise installation and Single-Sign-On.

5.3.6 Advanced Authentication

Berkeley Publisher does not offer Two-Factor Authentication.

5.3.7 Data Ownership

Berkeley Publisher can be installed on-premise or hosted in the cloud with negotiated Service Level Agreements, giving users various degrees of control over a document generation service. Templates and reusable parts of templates can also be managed locally.

6 dox42

dox42 is a document automation tool that runs in the cloud or is installed on-premise that is accessed for document generation only through a REST or SOAP API. It is complemented by a Microsoft Word Add-In that is primarily used to create templates but can also be utilized to manually generate documents.

6.1 Executive Summary

dox42 generates templates from external data sources or custom input fields. Excepts for multiple selectors, the tool supports all other input elements required in this study. Additionally, date values are manually typed by users instead of choosing from a date picker. Users can declare compound logical expressions with customized order of precedence, along with basic arithmetic operations.

Generating documents in dox42 starts with creating input fields or importing from data sources. Next, users place dynamic elements into their existing or new template. Finally, users trigger the document automation process, fill in data for created input fields, and dox42 generates a DOCX document containing inputted data in the template.

In general, the user interface of dox42 is minimalist. Configuring dynamic texts from input fields in dox42 is intuitive, yet constructing templates from data sources such as databases or XML files can be challenging. Users may need to analyze examples from dox42 to understand the underlying mechanism of data source configuration. dox42 offers validation for number and date/time values to prevent erroneous data of these types in the template generation process.

As a Word plugin, dox42 inherits collaboration, Two-Factor Authentication, users and access rights management features from Microsoft. In addition, dox42 server provides REST or SOAP interfaces for document automation on the server of users.

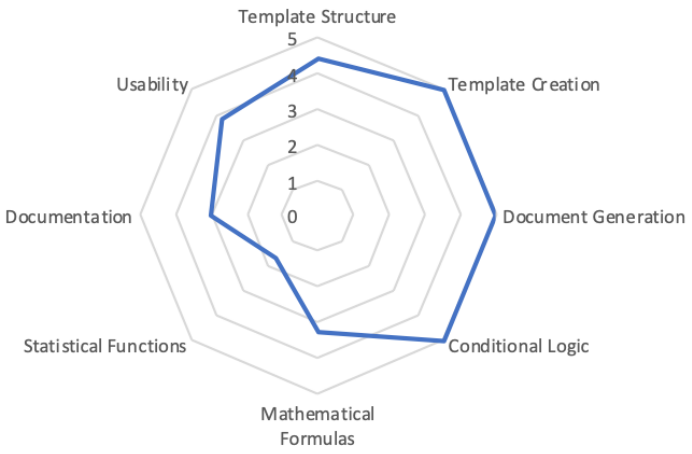

Figure 6.1: Radar diagram for dox42 - Core Document Automation Functionality

Integration	✓
Version Control	-
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	✓
○ Single Sign-On	✓
Data Ownership	
○ Cloud-based	✓
○ On-premise	✓

Table 6.1: Supporting features of dox42

6.2 Evaluation of Core Document Automation Functionality

6.2.1 Template Structure

Templates in dox42 consist of a list of user inputs and a document body. A set of user inputs can be used in any number of templates. User inputs are always of the data type string and can have a default value. dox42 supports three different types of user input:

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	-
Table	✓

Table 6.2: Template structure features of dox42

- ”Text”: Simple text input without formatting.
- ”Select from Values”: Dropdown menu with fixed options. Only one option can be selected at the same time.
- ”Select from Data Source”: Dropdown menu where the options are based on a column in a data source. Only one option can be selected at the same time.

The document body consists of formatted, static text in which references to user input values can be inserted. During the document generation process, these placeholders are replaced with the actual values. Words, phrases and text blocks inside the document body can be made conditional by attaching a conditional expression to them (see Section 6.2.4). The tool supports all the data types considered in this study, except for multiple selectors.

Score: 4.4/5

6.2.2 Template Creation

All parts of a template are created in a Microsoft Word Add-In and then deployed to the dox42 server. A new template can be based on an existing DOCX document or created from scratch. If there exists a list of user inputs, it can be attached to the current template. Users can also add new inputs from various external data sources.

The document body is filled with static text by users. References to user input values can be inserted using drag-and-drop. To make a text block conditional, users mark the text, apply conditional control over it, and declare logical expressions to display the text. Text blocks can be stored as DOCX files and can be reused in other templates. dox42 keeps track of the file locations of included text blocks from other templates. Changes can be pulled by refreshing all text blocks simultaneously.

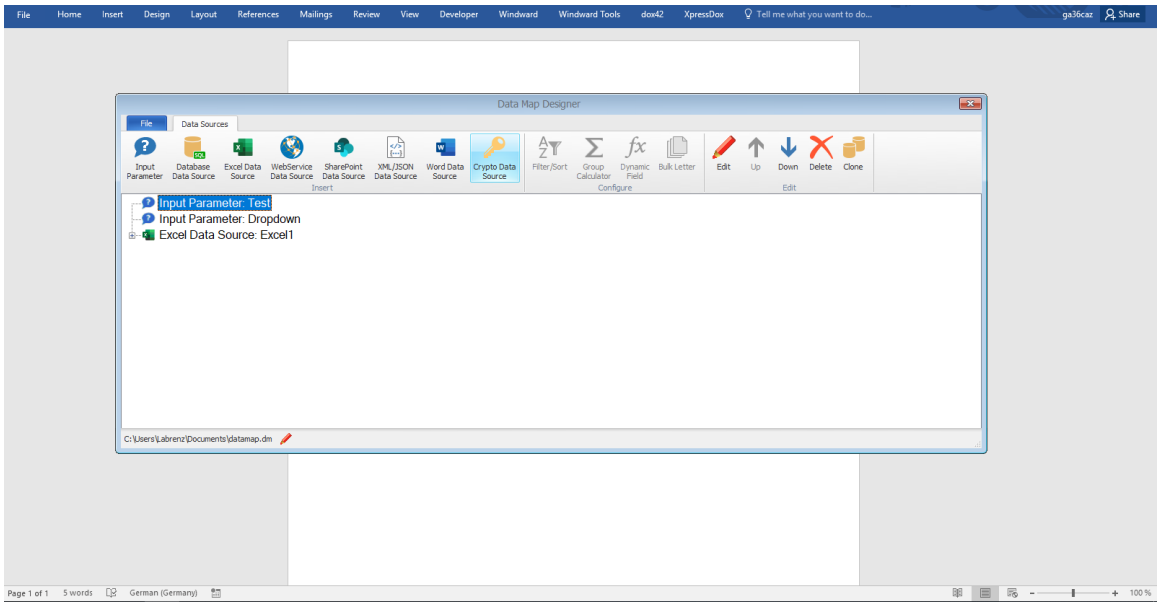

Figure 6.2: dox42 - Questionnaire Creation

Creating templates in dox42 is straightforward but all files created in the process have to be managed manually.

Score: 5/5

6.2.3 Document Generation

To generate a new document, dox42 first presents a set of questions to users. If there is at least one reusable template, users have to answer questions of this template on the next page(s). Templates can be configured to complement or substitute user inputs with external data sources like databases or XLSX files. After all questions have been answered, the document is generated and opened in a new window of Microsoft Word, where it can be edited further and saved.

The document generation process in dox42 is easy and can use multiple data sources if templates are configured appropriately. dox42 supports various connections to external data sources such as D365 CE/CRM, FO/AX and BC/NAV, MS SharePoint, Teams, SAP, SQL, WebService, and XML/JSON. Besides DOCX documents, the tool can also generate spreadsheets and presentation slides.

Score: 5/5

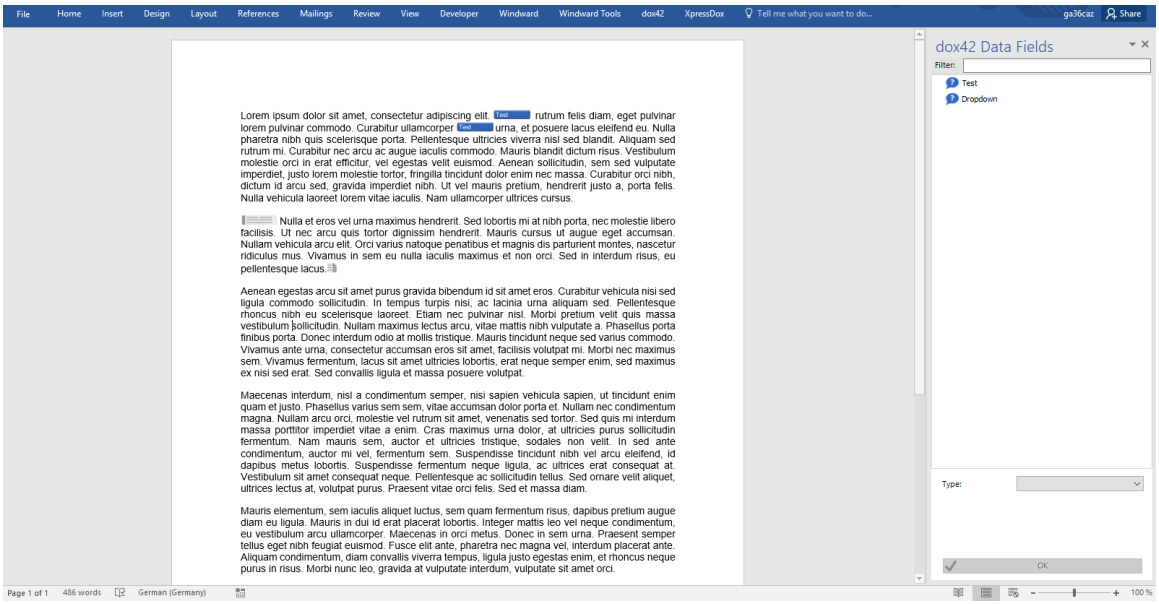

Figure 6.3: dox42 - Document Body Creation

6.2.4 Conditional Logic

Conditional expressions can be applied to words, phrases and text blocks to decide if they are displayed or hidden. User input prompts during the document generation cannot be made conditional. A conditional expression consists of any number of comparisons of user input values with other inputs or static values. The comparisons can be nested and chained together using any combination of brackets and the operators ”AND”, ”OR” and ”NOT”. The available operators for comparisons are =, <>, >=, <=, < and >.

Conditional expressions are not validated during the template creation and will result in a failed document generation if a syntax error occurs. Error messages are generic, displaying only the conditional expression that resulted in an error, but not at which position in the document the expression was placed.

Conditional logic in dox42 is flexible and can model any complex logical expression, but does not offer a feature to validate conditional logic during the template creation process.

Score: 5/5

6.2.5 Formulas

Formulas in dox42 can be applied to external data sources and user inputs. Available functions are based on the Visual Basic programming language. The formula is automat-

ically validated and syntax errors are highlighted in the code editor. Additionally, certain statistical functions like ”sum”, ”avg”, ”min”, ”max” and ”count” combined with filter expressions can be applied to columns in external data sources.

Basic arithmetic and certain advanced formulas in dox42 are applicable to both data sources and user inputs. However, mathematical formulas are applied to user inputs using Excel mathematical functions. Therefore, users need to create a spreadsheet to apply mathematical calculations on user inputs, and consequently extract the results of the calculations as input fields.

Basic and Advanced Formulas	:	**3.3/5**
Statistical Functions	:	**1.7/5**

6.2.6 Documentation

The documentation for dox42 consists of a list of tutorials, example templates and manuals. A manual that describes the Add-In is not available. The documentation is aimed at non-technical users and contains many videos and examples, including sample templates for users to understand how to use dox42 features.

Score: 3/5

6.2.7 Usability

The user interface of the Add-In is clean and easy to use with dialog windows only displaying the necessary amount of information and important features that are easy to find. This makes the document creation process very fast. Highlighted references to user inputs and conditional text blocks help to keep an overview of the template. This becomes worse during the document generation of large templates when the questionnaire based on the required user inputs displays only a single question at a time.

Score 3.8/5

6.3 Evaluation of Supporting Features

6.3.1 Integration

The dox42 server provides a REST and SOAP API to generate documents from templates. Additionally, document automation features can also be integrated into Microsoft D365 and O365, SAP, K2, Nintex, FireStart, d3, or DocuSign.

6.3.2 Document Management

dox42 does not offer a document management system. However, users can leverage a version control feature from external cloud-based file management systems.

6.3.3 User Management

dox42 does not provide integrated user management, however, as a MS Word Add-In, the tool provides possibility for users to leverage the template sharing and more advanced collaboration features provided by Microsoft.

6.3.4 Collaboration

Support for collaboration between users is limited because dox42 does not provide user management and communication, therefore teamwork has to happen outside of the tool. If documents are manually generated from the Add-In, the necessary templates have to be shared manually.

6.3.5 Enterprise Scalability

It is possible to scale dox42 for large organizations by deploying it on a company server.

6.3.6 Advanced Authentication

As a Word plugin, Two-Factor Authentication is provided to users via a Microsoft Authenticator Mobile app to access Word documents and use the dox42 plugin afterwards.

6.3.7 Data Ownership

dox42 can be installed on-premise or run in a public cloud. All templates and documents are managed outside of the engine and are stored in the DOCX file format.

7 Lawlift

Lawlift is a web-based document automation tool and only runs on a public cloud. It is developed by the startup with the same name since 2017.

7.1 Executive Summary

Lawlift follows a minimalist approach in the design of its user interface. Except for numeric input fields, all other data types are supported. Although Lawlift supports complex logical expressions, they are only applicable to single and multiple selections. Besides, mathematical formulas cannot be included in logical expressions.

To automate documents in Lawlift, users create questions and specify positions of their answers in a template. Next, users fill in the questionnaire and Lawlift generates a DOCX document in return.

Questionnaires in Lawlift cannot be created directly in a template, but users have to create them as separate modules, and include these modules into the main template. The validation mechanism of the tool is limited so unexpected inputs are not completely filtered when generating documents.

The tool offers version control, user groups and access rights management features. Lawlift also provides APIs for its document automation features to external systems. Noticeably, Lawlift provides security protections by encrypting data entered when generating documents on the client side, and hosting the application on server centers being certified with ISO 27001 standard.

In conclusion, Lawlift is a minimalist document automation tool which does not support complex logical expressions or mathematical formulas.

7.2 Evaluation of Core Document Automation Functionality

7.2.1 Template Structure

A template in Lawlift consists of a questionnaire and a document body. Questions, paragraphs and single words can be made dependent on conditional logic (see Section 7.2.4).

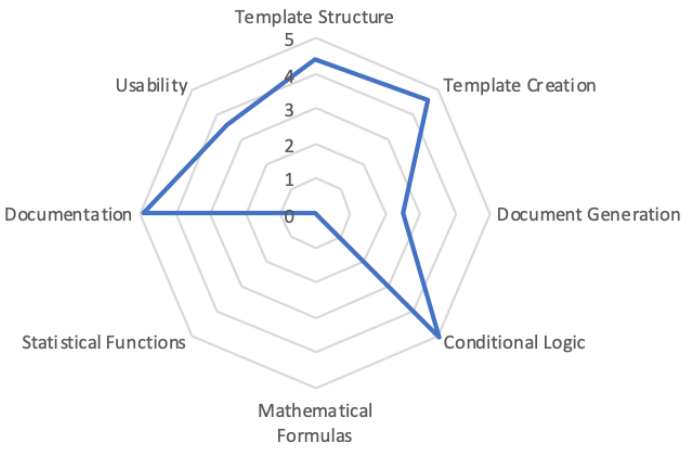

Figure 7.1: Radar diagram for Lawlift - Core Document Automation Functionality

Integration	✓
Version Control	✓
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	✓
◇ via Mobile App	-
○ Single Sign-On	-
Data Ownership	
○ Cloud-based	✓
○ On-premise	-

Table 7.1: Supporting features of Lawlift

The questionnaire consists of any number of questions split among multiple sections. The supported data types for questions are listed in Table 7.2:

The document body consists of headings containing any number of paragraphs made up of static text and references to questions. A limited set of formatting options is available that can be applied to paragraphs.

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	-
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 7.2: Template structure features of Lawlift

Templates in Lawlift follow a minimalist approach, while supporting most of the required data types, except for numeric input.

Score: 4.4/5

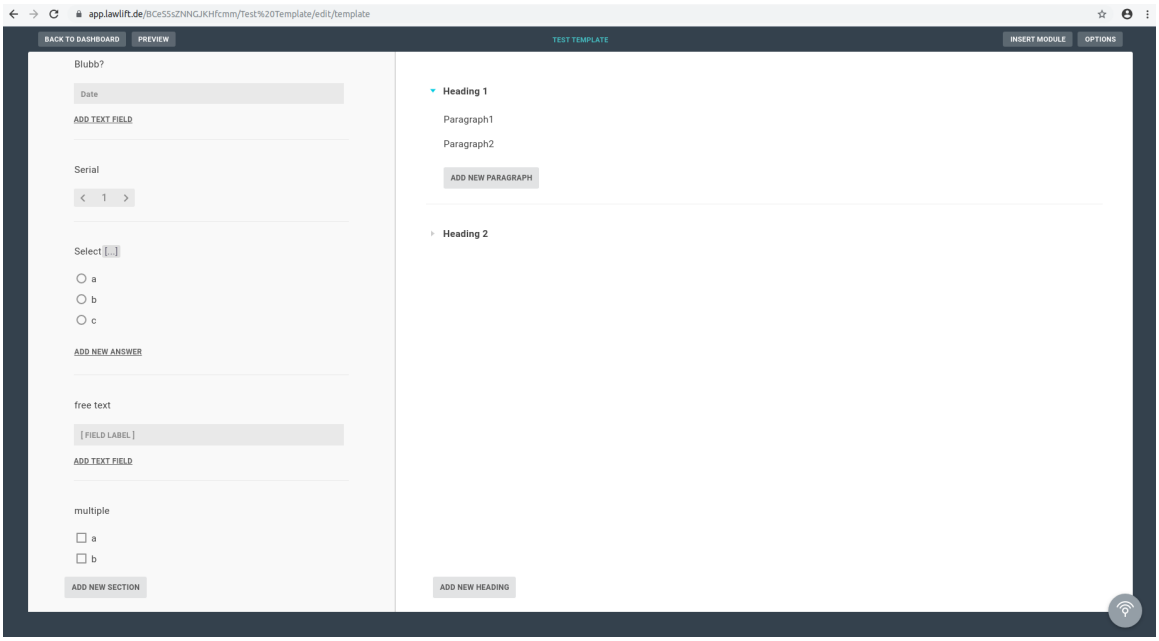

Figure 7.2: Lawlift - Template Structure

7.2.2 Template Creation

Templates are created in a web-based editor where the questionnaire and the document body are shown next to each other. Elements, questions, and paragraphs can be added, rearranged and made conditional. The structure of the questionnaire and document body can also be modified. Paragraphs contain static text where users can insert references to the answers of questions.

Importing documents from files is not possible, all templates have to be created from scratch. Users can create reusable modules, store them in a repository in the cloud and insert them into other templates. Reusable modules are structured like templates consisting of a questionnaire and a document body. They are managed centrally and can be updated in all templates where they are used simultaneously. Values of answers to questions in a reusable module can be synchronized for all documents that have been classified as belonging to the same client or case.

Changes made to a template are saved automatically and can be used immediately for generating new documents.

Overall, template creation in Lawlift is smooth but lacks document import and version control for templates.

Score: 4.6/5

7.2.3 Document Generation

During the document generation process all questions are shown at the same time and can be answered in any order. A preview of the document based on the given answers is shown next to the questionnaire. The process can be paused and resumed afterwards. Users can download the generated document as a DOCX file at any time. The document generation process can always be continued to generate and download a new version of the document.

User inputs cannot be substituted with a database or a file upload and documents cannot be generated using batch processing.

The document generation process is streamlined for a simple download of a single DOCX file, other applications are not supported without accessing the API.

Score: 2.5/5

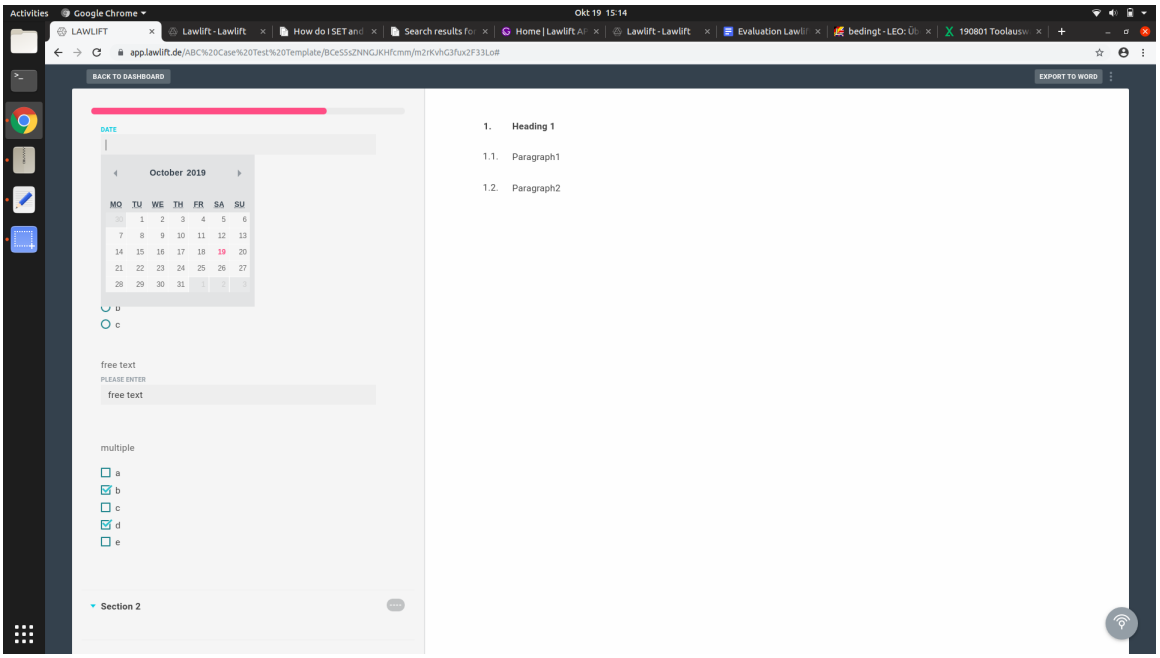

Figure 7.3: Lawlift - Document Generation

7.2.4 Conditional Logic

Conditional logic can be used to decide if a question, paragraph or single word is displayed. A conditional rule consists of any number of conditional expressions that can be chained together by ”AND” or ”OR” operators. A conditional expression is only applied to options of the ”Select” or ”Multiple” question type. Users can specify if either all options have to be selected or at least one option has to be selected for the condition to be evaluated as true. Questions with the data type ”Text” or ”Date” cannot be used for conditional expressions.

Lawlift supports declaration of compound and grouped conditional logic for most frequent use cases, but it is restricted to the Multiple Selector question type.

Score: 5/5

7.2.5 Formulas

Lawlift currently does not provide support for mathematical calculations.

Basic and Advanced Formulas	:	**0/5**
Statistical Functions	:	**0/5**

7.2.6 Documentation

The documentation is aimed at non-technical users and is available online for the public. It is well-structured, searchable, and consists of a large number of articles that describe all relevant features in detail with illustrative examples and videos.

Score: 4.9/5

7.2.7 Usability

The user interface of Lawlift is intuitive and minimalist resulting in a short time to get familiar with the tool. All important functionalities can be accessed from a central navigation menu and a dashboard. Besides, Lawlift provides a mechanism to undo and redo actions while editing questionnaires and templates.

Lawlift lacks a mechanism for input validation, for example users can mistakenly insert text instead of numeric values in a date field, but the tool still accepts the erroneous input.

Score 3.6/5

7.3 Evaluation of Supporting Features

7.3.1 Integration

Lawlift provides a RESTful API with endpoints to list all available templates and to generate a document from a template.

7.3.2 Document Management

Lawlift does not provide a document management system but keeps track of previously generated documents that can be changed and downloaded again.

7.3.3 User Management

User management in Lawlift is very flexible with customizable user roles and user groups. Administrators can invite and manage users and user groups. User roles can be defined freely and are attached to a set of very granular permissions for every functionality. User roles are assigned to individual users and each user can have any number of user roles. User groups can share templates and generated documents.

7.3.4 Collaboration

Users can share templates and generated documents with user groups allowing teams to draft multiple versions of a document in collaboration.

7.3.5 Enterprise Scalability

Lawlift's powerful user management allows even large organizations to map user roles to their needs. Managers and administrators can analyze the document automation process in the organization by utilizing a central metrics dashboard.

7.3.6 Advanced Authentication

Lawlift enhances security by providing Two-Factor Authentication via email for accessing documents with a security code.

7.3.7 Data Ownership

Lawlift is a cloud-based tool that cannot be installed on premises. Moreover, it cannot run locally in a web-browser without uploading user inputs to the cloud, giving users limited control of the document automation service and their data. The service is hosted in Germany and certified with ISO 27001 standards to secure information asset. Templates and reusable modules can be downloaded as JSON file and shared offline.

8 Legito

Legito is a cloud-based document automation tool that can also be installed on premises. It is developed by the company with the same name that started out as a provider of pre-made document templates in 2014. Since then it has become a major player in the document automation market.

8.1 Executive Summary

Legito offers an intuitive drag-and-drop method to create templates for document automation. Nevertheless, users must follow the hierarchy defined by Legito to add content into their templates. Most input types considered in this study are supported by the tool, except for text area, i.e. multiple line text input. Legito provides an intuitive yet powerful selector-based interface to define and group logical expressions. Additionally, users can also apply basic and advanced mathematical operations to dynamic elements and logical expressions in the template.

The process of generating documents from templates is comprised of three steps: compose template, define dynamic inputs along with their questions, and fill in the questionnaires to generate documents in PDF or DOCX format.

It is possible to define user groups and set their read or write access in Legito, enabling the collaboration between different parties during the document automation process.

Regarding usability, the user interface of Legito is minimalist and intuitive, users can easily access all relevant features to accomplish their tasks. There exist several glitches in the template editor view, causing uncomfortable user experience while composing templates. Additionally, Legito does not offer flexible styling format for text editing. Each text element in Legito is assigned a predefined font size, so users can only apply bold, italic, or underline formatting.

Legito provides many functionalities that are important for document automation and can be used as a cloud-based service or installed on premises if organizations have strong data protection requirements.

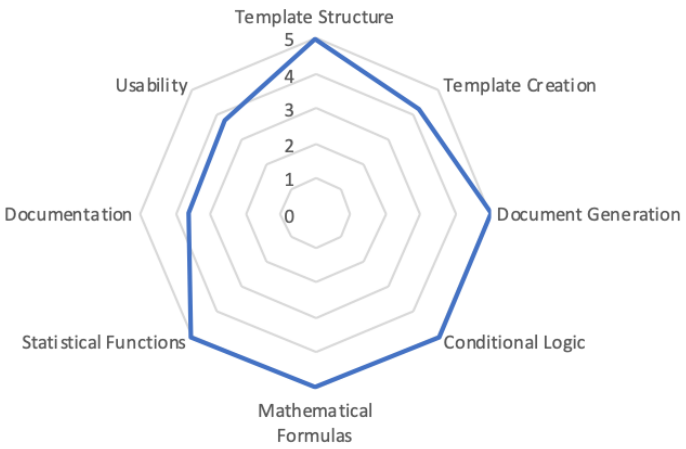

Figure 8.1: Radar diagram for Legito - Core Document Automation Functionality

Integration	✓
Version Control	✓
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	-
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	-
○ Single Sign-On	-
Data Ownership	
○ Cloud-based	✓
○ On-premise	✓

Table 8.1: Supporting features of Legito

8.2 Evaluation of Core Document Automation Functionality

8.2.1 Template Structure

The structure of a template in Legito is usually divided into three parts: header containing information like contractual parties or addresses, document body and footer. The document body consists of any number of hierarchical clauses. The hierarchy from top to

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 8.2: Template structure features of Legito

bottom is article paragraph, sub-paragraph, point and item. The template structure can be changed on every level by drag-and-drop. Each clause can contain multiple elements in any order.

The document structure is automatically numbered correctly but numbering can be turned off for single articles or the whole document. A template can contain multiple documents that are generated at the same time. References and conditions can span across all documents inside a template. Style templates can be uploaded and used in templates to automatically format generated documents with the corporate design.

Legito templates are very well structured and support most important data types, except for multiple selector and text area.

Score: 5/5

8.2.2 Template Creation

When a new template is created, it can be created from scratch or imported from a DOCX file or an existing Legito template. If a DOCX file is imported, Legito translates the document to its hierarchical structure of clauses. This process saves time but still requires a number of manual adjustments. Inserting or rearranging clauses and elements is done using the drag-and-drop mechanism. Legito ensures that the internal structure is adhered to and highlights possible positions for a clause and element.

Users can specify which clauses of the template can be changed and if additional clauses can be appended manually after the document automation process. Formatting can be applied to the whole template, clauses or text fields. Articles can be copied from other templates to reduce rework but changes in the original template do not affect their copies.

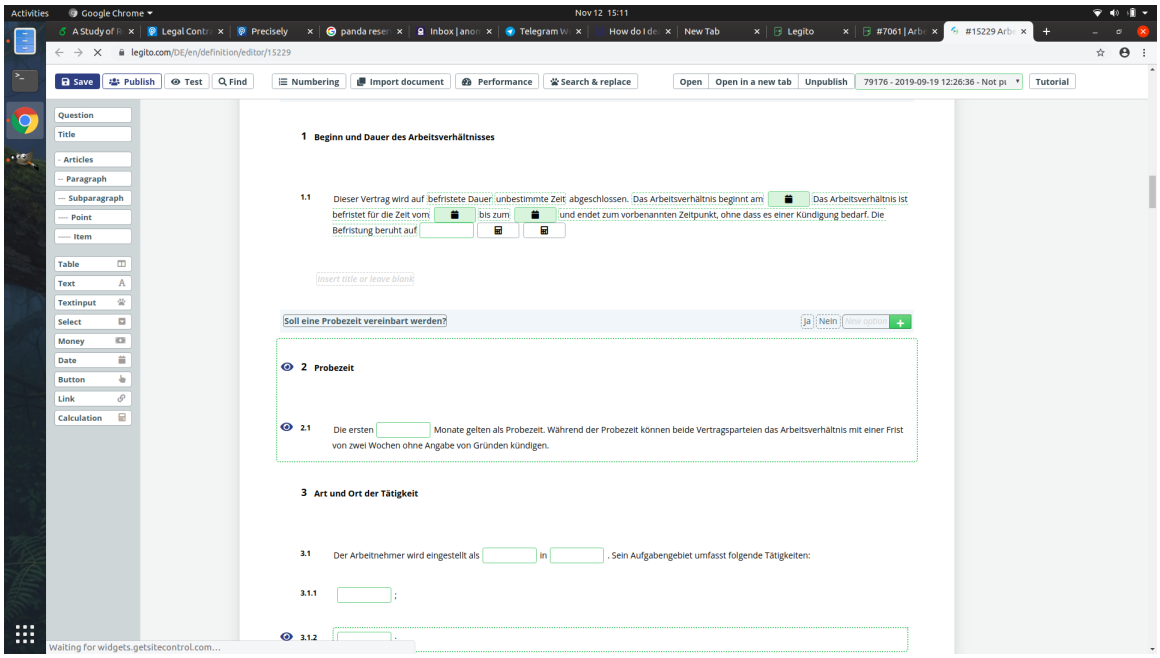

Figure 8.2: Legito - Template Structure

After changes have been made to a template, it has to be saved manually. Old versions of a template can be accessed, edited and used to generate documents. The user can look at a preview of the latest saved version and publish the template to make it available for other team members.

Creating templates in Legito is fast and intuitive thanks to the drag-and-drop system. Although templates are highly reusable, users must follow the hierarchy of Legito when drafting their documents.

Score: 4.2/5

8.2.3 Document Generation

During the document generation process, the user is presented with a simplified copy of the template and enters values into user input elements. Text blocks can be changed and clauses can be added if the template's settings allow it. Each clause and element can be annotated with a help text. Warning messages can be attached to text input elements to validate numeric values.

Users can save progress on the documents, download or send them via email. In addition to DOCX and PDF Legito supports export as HTML, RTF, XML, ODT and TXT files. If

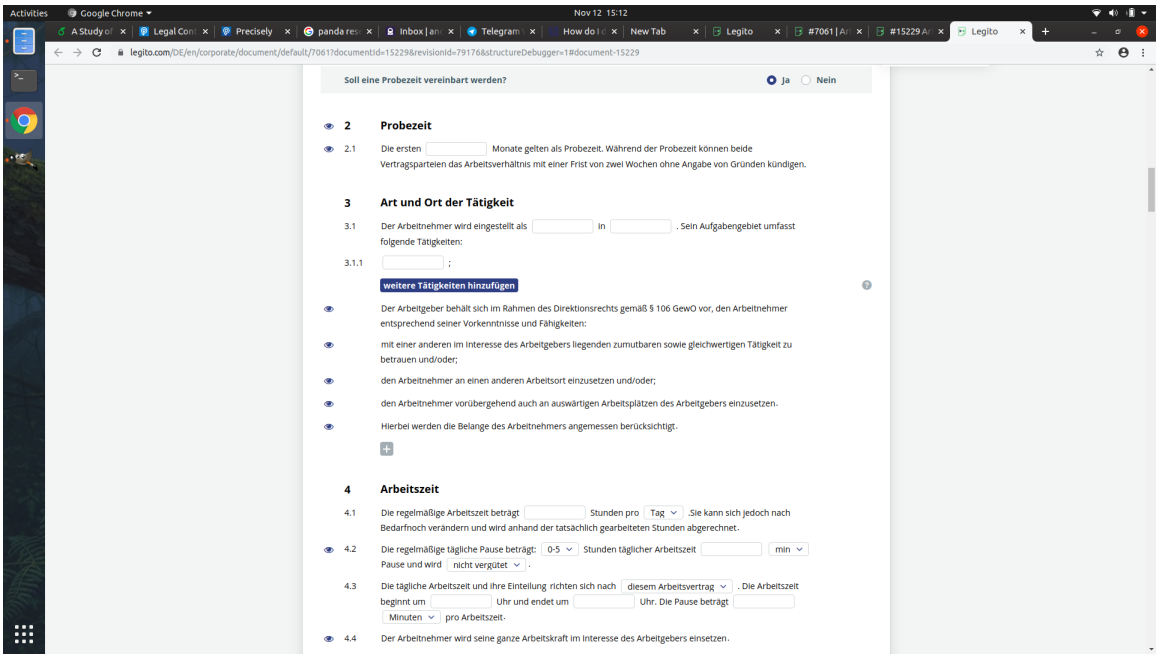

Figure 8.3: Legito - Document Generation

a template consists of multiple documents they are generated and downloaded together. A database cannot be integrated to substitute manual user inputs but XLSX and CSV files can be used to generate documents in bulk.

Legito offers many options to customize the generated document and supports a wide variety of file formats but is not able to use a database as data source.

Score: 5/5

8.2.4 Conditional Logic

Each clause, element and additional document of a template can be linked to a conditional expression that decides if it is displayed or hidden. A conditional expression consists of one or many comparisons of an element's or clause's value with a static value. Comparisons can be grouped or chained together using multiple logical "AND" or "OR" operators. Available operators for comparisons depend on the element's data type and contain =, !=, <, >, <=, >=, "is visible", "is not visible", "is empty", "is not empty", "is set to", "is not set to", "is turned on" and "is turned off".

Legito's conditional logic can be applied to every part of a template and can model all frequent use cases.

Score: 5/5

8.2.5 Formulas

Available operators and functions for formulas are based on the PHP programming language and are not explained inside the tool. Values of elements can be included in a formula by referring to their system name. Formulas are only validated during the document automation process and can only be debugged by the customer support.

Although it is possible to define statistical functions in Legito, users need to know JavaScript in order to write these functions as Legito tags. This is challenging for non-technical users who do not have web programming knowledge.

Legito supports all mathematical operations considered in this study, but a lack of explanations and guidance makes them hard to use by new users.

Basic and Advanced Formulas	:	**5/5**
Statistical Functions	:	**5/5**

8.2.6 Documentation

The documentation is aimed at non-technical users and consists of a blog, video tutorials, feature documentations and a training book. There is no centralized structure or search function for the documentation.

Score: 3.6/5

8.2.7 Usability

Legito is completely cloud-based, making setup effort trivial. The user interface has an intuitive design, looks modern and hides the tool's complexity well. All important functionalities can be accessed from a central navigation menu and a dashboard.

Additionally, Legito offers quality-of-life features for template creation like document-wide search-and-replace, full-text search or analyzing a template's performance during the document automation process.

Creating templates with only drag-and-drop while automatically enforcing document structures provides a positive user experience. However, to arrange the content of their templates, users must follow the structure defined by Legito. This makes the process of creating template structures in Legito not flexible.

In terms of user control and freedom, Legito does not provide an option to undo or redo editing actions. Furthermore, in general, there are no error messages to inform users about the source of error. For instance, when dragging an invalid component inside a template, users are not informed why Legito does not accept it.

Score 3.7/5

8.3 Evaluation of Supporting Features

8.3.1 Integration

Legito provides a RESTful API with endpoints for all important functionalities like managing templates, generating documents from templates and managing users.

8.3.2 Document Management

Central element of Legito's document management system is a filterable list of all documents that the user has generated or has been permitted to see. Documents can be downloaded, edited, updated to a new version, shared, deleted and anonymized by automatically removing confidential data. Documents can have a variety of attributes like monetary value or due date that can be set during the document generation process. Related documents can be uploaded and linked. Document records can be extended and shrunk to display different degrees of detail.

8.3.3 User Management

Administrators can invite team members and manage permissions to use and edit templates for single users or user groups. Each document in Legito has an owner with full access that can grant permissions to users or user groups. Available permissions are:

- View record only: The user can see that the document exists
- View document: The user can view the document but can't make any changes
- Edit document: The user can make changes to the document
- Manage document: The user has the same permissions as the owner

Legito's user management allows all permissions to be set at many different levels of granularity.

8.3.4 Collaboration

Users can share documents with team members or the whole organization with different levels of access (see Section 8.3.3). They can view different versions of shared documents, create new versions and add comments.

Support for teamwork on documents is very intuitive and allows all relevant communication to go through Legito.

8.3.5 Enterprise Scalability

Legito scales very well both integrated with other systems or as a standalone tool. It can handle huge amounts of templates, documents, and users with permissions that are manageable on any granularity. Enterprise customers can expect technical support 20 hours a day and a customer success manager to help with strategic decisions.

Legito combines a horizontally scalable tool with services geared towards enterprises to ensure a high-value realization.

8.3.6 Advanced Authentication

Legito does not support Two-Factor Authentication and Single Sign-On features.

8.3.7 Data Ownership

Legito is a cloud-based tool by default but can be installed on premises on demand if an organization's data is very sensitive. In this case, the customer retains full control of the software and its data.

9 Logiforms

Logiforms is a cloud-based data collection tool utilizing online user input forms. Being developed since 2001, Logiforms provides an interface to create responsive web forms for collecting, storing, and distributing input data to internal and external sources. Logiforms also offers a feature to create PDF documents from input data and send them via email as attachments.

9.1 Executive Summary

Logiforms supports all input types of the defined evaluation criteria. Noticeably, users can customize validation and define masking for particular input types. Furthermore, users can apply conditional and arithmetic operations on these inputs to customize a template's structure.

Regarding usability, Logiforms offers user-friendly and straightforward methods to construct components for questionnaires. Users can construct multiple pages for their set of questionnaires. The overall graphical design is minimalist, with frequently used features being placed at an easily accessible position, and all extra component properties being compactly displayed in a small area.

To create a dynamic template from input data, Logiforms presents an editor for users to type and style the content of their text and form inputs. The template structure is invoked to create a PDF file after the form is submitted. The generated PDF file can be attached to an email, or kept in Logiforms's database.

Although users do not need advanced technical skills to create simple input forms and generate documents, experience with software development or spreadsheet programs significantly reduces effort to customize input properties, logical conditions and troubleshooting.

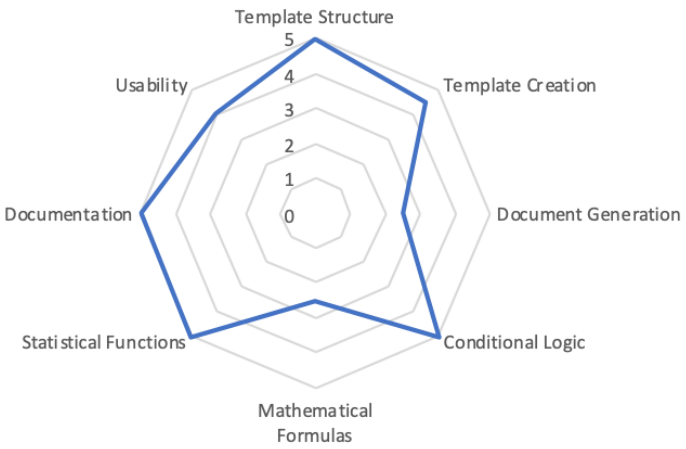

Figure 9.1: Radar diagram for Logiforms - Core Document Automation Functionality

Integration	✓
Version Control	-
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	-
○ Single Sign-On	-
Data Ownership	
○ Cloud-based	✓
○ On-premise	-

Table 9.1: Supporting features of Logiforms

9.2 Evaluation of Core Document Automation Functionality

9.2.1 Template Structure

A template in Logiforms consists of a questionnaire and one or more documents that are generated at the same time. A questionnaire consists of any number of elements that can be split among multiple pages. Each element is identified by a name and can be assigned

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 9.2: Template structure features of Logiforms

an automated task to modify values in other places or execute custom JavaScript code. There are more than 40 different types of elements available with eight elements that are most relevant for this evaluation are illustrated in Table 9.2.

In the questionnaire and template editor, elements can be hidden or enabled based on conditional logic (see Section 9.2.4) and structured into panels and groups. Elements with user input can have a default value, contain a formula to store a calculated value (see Section 9.2.5) and be validated during the document generation process.

A template structure is composed with an editor, in which users define and format the content of the generated document. The template can contain both static text, and referenced values of elements in the input form.

Score: 5/5

9.2.2 Template Creation

Questionnaires and documents in templates are created separately with documents requiring elements in the previously created questionnaire. Elements can be added to questionnaires by clicking on the *insert element* button on the toolbar, or right-clicking on the element tree or main interface. To rearrange elements in questionnaires, users drag an element and drop it in the desired position in the Form Outline Tree on the left. Elements can be annotated with notes that only template creators can see to explain their structure.

Elements and groups of elements from an existing template can be saved to reuse them in other templates. After changes have been made to a template it has to be saved manually and published for the changes to take effect. Old versions of templates cannot be accessed or used.

Documents are generated as part of the post-processing of a questionnaire. They are stored as PDF files in Logiforms's cloud and appended to the questionnaires. A new document can be created from scratch or imported from an uploaded DOCX file which is then converted to HTML. Users can insert static texts and references to elements from the questionnaire which can be constrained or manipulated by user-defined conditions. Documents are formatted in the Logiforms editor and the underlying HTML code can be changed by advanced users.

Users have many options for creating and editing templates. Templates can be managed centrally in a repository but without version control.

Score: 4.5/5

9.2.3 Document Generation

To generate documents from a template, users have to answer questionnaire(s) first. Elements of the questionnaire can be annotated with help texts to guide the users. After users submit their answers to the questionnaire(s), multiple post-processing steps can be triggered, most importantly the creation of a PDF file from a document or customized template. To access the generated PDF documents, users click on the *Database button* on the toolbar, double-click on a desirable record, and scroll down to the bottom of a *Record Detail* window where the PDF files are listed. Other file formats including DOCX are not supported. Multiple documents can be generated simultaneously and can be merged into a single file. User inputs can be substituted by a previously imported XLSX or CSV file but not by data imported from a database.

Generating documents is one of the several available post-processing steps after users answer a questionnaire. Another feature of Logiforms is sending the generated documents via emails. However, Logiforms cannot generate DOCX documents, so users can only create PDF files from templates.

Score: 2.5/5

9.2.4 Conditional Logic

Elements in questionnaires and their references in templates can be displayed or hidden based on conditional logic. Users can mark these elements as "required" for the validation step during the document automation process.

A conditional expression consists of one or many comparisons of the value of an element with a static value or another element's value. The tool supports compound logical ex-

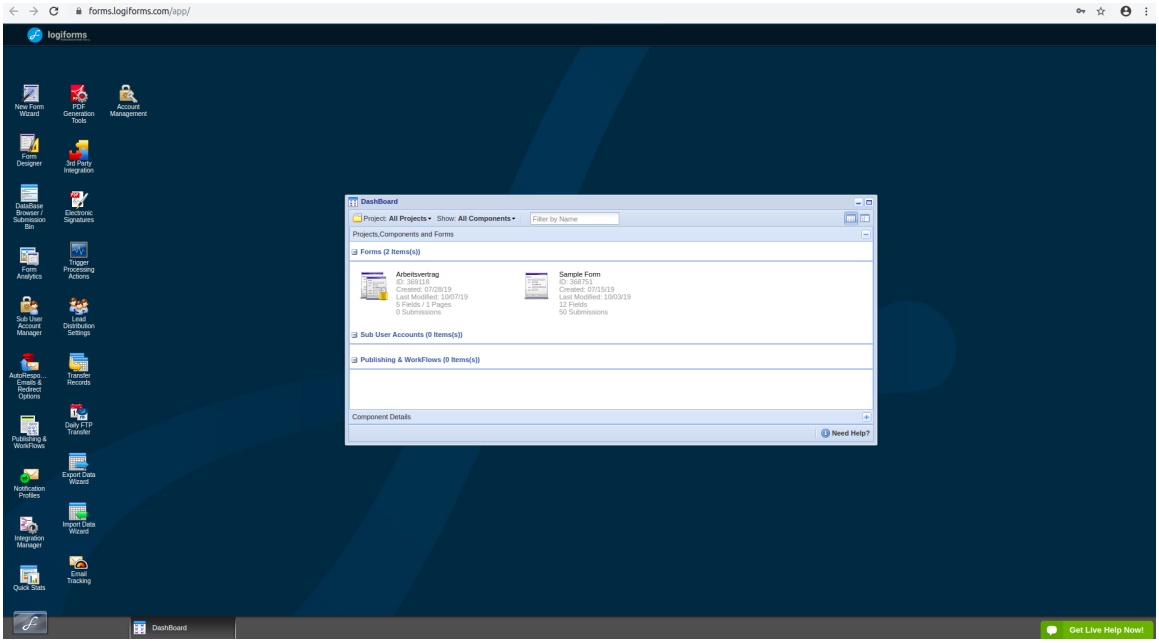

Figure 9.2: Logiforms - Main Screen

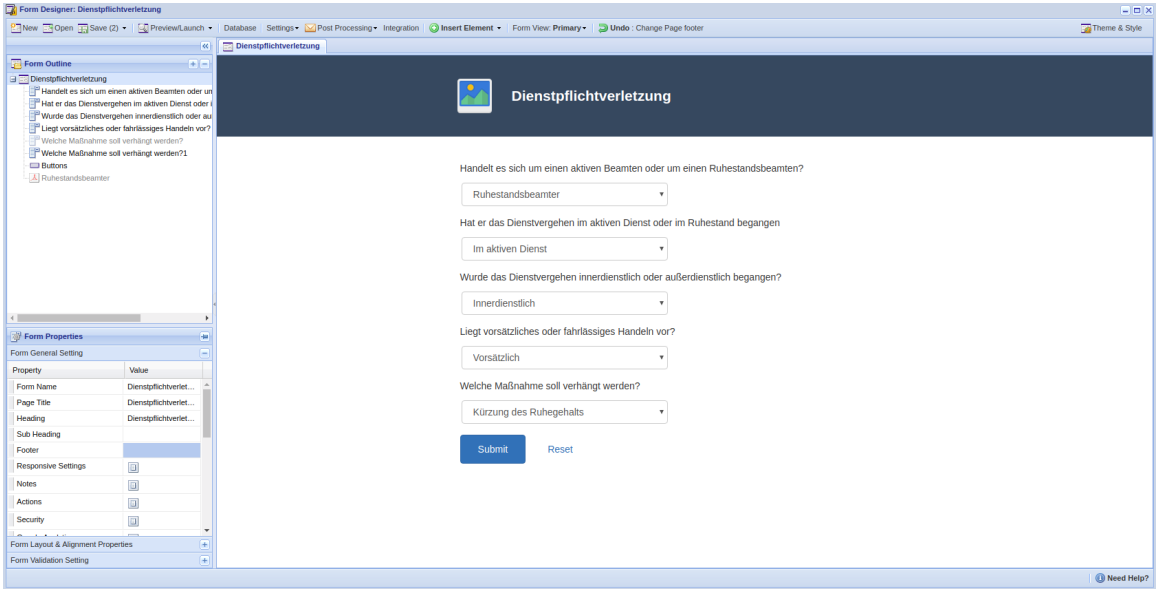

Figure 9.3: Logiforms - Template structure

pressions with customized order-of-precedence.

Logiforms presents a straightforward interface for users to select or define elements in their logical expression. Besides, advanced users can directly write their expressions with free-text and mathematical formulas (see Section 9.2.5). It is possible to construct nested conditional logic by using any combination of logical "AND", "OR" and "NOT" operators, and brackets. The syntax of conditional expressions is not validated during its creation. Available operators for comparisons are =, !=, <, >, <=, >=, "is empty", "is not empty", "contains", "does not contain", "starts with", "does not start with" and "in (for lists)".

To summarize, Logiforms provides a very powerful and flexible mechanism to process conditional expressions at level 4 complexity (see Section 2.1.4) containing mathematical operations. However, while writing logical expressions in free-text, it is not possible to undo a typing mistake.

Score: 5/5

9.2.5 Formulas

Mathematical calculations can be applied to documents inside templates, conditional expressions and elements in questionnaires. Logiforms supports basic arithmetic operations along with several advanced mathematical functions such as exponent and square root. Formulas are only validated during the document generation process. They can be used flexibly and support all necessary operators but have to be validated manually.

Basic and Advanced Formulas	:	**2.5/5**
Statistical Functions	:	**5/5**

9.2.6 Documentation

The documentation is very detailed but understandable by non-technical users. It consists of structured and searchable descriptions of template creation and modification, generating documents, using conditional expressions and mathematical formulas features. Logiforms also illustrates the features' usage with examples and video tutorials.

Score: 5/5

9.2.7 Usability

Among the advantages of Logiforms is the constant update about the outcomes from users' interaction. Additionally, the language and position of elements in graphical design is consistent across different user interfaces. Logiforms has efficient mechanisms to prevent user

errors. For example, various input validation options suspend the submission of unexpected data; logical expression containing syntax errors, invalid logical expressions with syntax error, undefined referenced value or mathematical formulas are not accepted. Finally, Logiforms has a detailed, concise and illustrative documentation to help users understand how to use system features.

A number of improvements can be made regarding the workflow of template generation. Several user interfaces are intuitive, especially for displaying generated documents, creating or reusing templates. For instance, if there exists a button to view generated reports when users select a record in the Database view, then one can recognize the possibility to view the generated file inside each record. Aside from that, generating templates by merging PDF or DOCX documents is a flexible feature. Nevertheless, the procedure for merging other dynamic templates is not as simple as referencing to the template sources, and the questionnaire for that template will be presented to users. Instead, users need to be familiar with the Subform feature of Logiforms to automatically reuse templates.

One noticeable inconvenience of defining conditional logic is that users cannot undo typing mistakes. Furthermore, error messages for invalid logical expressions are not informative enough for users to locate the source of error.

Score 4/5

9.3 Evaluation of Supporting Features

9.3.1 Integration

Logiforms provides a RESTful API with endpoints for functionalities displaying templates or updating documents that have been generated from templates. Managing templates and generating documents from templates using API calls is not possible.

9.3.2 Document Management

Logiforms saves user inputs for questionnaires in a dedicated database from which documents are generated. The tool also offers a system to manage questionnaires and templates. There exists no version control system in Logiforms.

9.3.3 User Management

Administrators can create additional user accounts and set permissions to manage templates and generated documents. Permissions to generate documents from a template cannot be set from within the tool.

9.3.4 Collaboration

Templates and documents generated from templates cannot be shared by users and have to be made accessible by an administrator. Documents cannot be changed manually once they have been generated but user inputs to the underlying questionnaire can be changed by team members if they have sufficient permissions triggering the creation of a new version of the document.

Support for collaboration in Logiforms is limited to updating documents that have been previously generated by a team member.

9.3.5 Enterprise Scalability

Questionnaires can be included into any website by copying the underlying HTML code allowing for an easy integration into the existing system environment. Customer support by email and chat is available 24/7 but there is no additional technical support for enterprise customers. While templates and their access policies can be managed centrally, generated documents have to be handled manually resulting in additional organizational effort.

9.3.6 Authentication

Logiforms does not support Two-Factor Authentication and Single Sign-On features.

9.3.7 Data Ownership

Logiforms is a cloud-based tool that cannot be installed on premises. Each organization has a completely isolated environment in the public cloud with their own database that holds all data related to generated documents. These data can be downloaded at any time but templates can only be accessed in the cloud.

10 Nintex Workflow Cloud

Nintex Workflow Cloud (in the following called Nintex) is a tool for workflow automation that runs on a public cloud. Document automation is only one of the many functionalities that can be triggered in an automated workflow. The tool is developed by Nintex since 2006.

10.1 Executive Summary

Nintex is a powerful process management tool and document automation is only one of its features. Generating templates in Nintex requires certain technical knowledge and experience with cloud storage platforms such as Google Docs or Dropbox. All of the input types considered in this survey are supported by Nintex. Logical expressions and mathematical formulas are also provided to customize the content of templates.

Before generating documents, users need to create a template with static content in a cloud storage platform. Next, users create questionnaires containing dynamic input fields. Each field is automatically assigned with a variable name by Nintex. To include the dynamic inputs into templates, users copy the variable names from Nintex to specific positions inside the templates. After that, users create a workflow task of generating documents, specify the template location in a cloud storage platform, save the decision tree and publish it. Finally, a link to the questionnaire is returned to the users. Filling and submitting this form will generate a new document inside the configured location of the user's cloud storage.

In terms of usability, the processes of generating documents, defining logical conditions with mathematical formulas using decision trees of workflow tasks, and configuring the storage location of generated files can be uneasy to non-technical users in the first attempts. However, since Nintex has an intuitive drag-and-drop mechanism and minimalist user interface, the document automation procedure becomes less challenging over time. Another concern is that Nintex takes about one minute to generate documents after submitting questionnaires.

The tool offers APIs for external systems to utilize its features. Sharing documents to other users and setting their access right is possible, but Nintex does not support creating user groups.

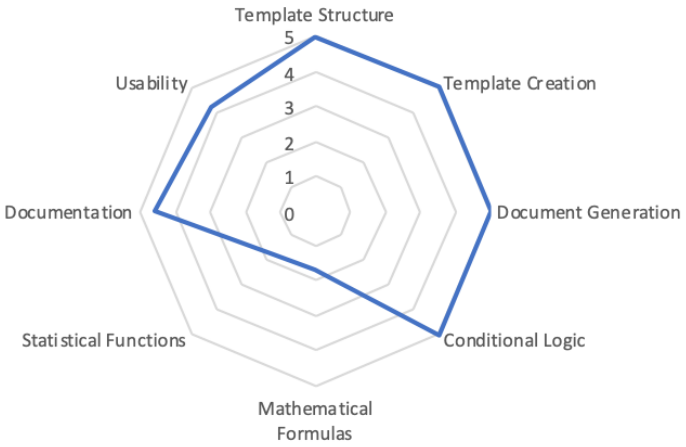

Figure 10.1: Radar diagram for Nintex - Core Document Automation Functionality

Integration	✓
Version Control	-
User Management	
○ User Group Management	-
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	-
○ Single Sign-On	✓
Data Ownership	
○ Cloud-based	✓
○ On-premise	-

Table 10.1: Supporting features of Nintex

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 10.2: Template structure features of Nintex

10.2 Evaluation of Core Document Automation Functionality

10.2.1 Template Structure

Constructing a template in Nintex requires an automated workflow, a questionnaire and a document body. A questionnaire consists of any number of questions that populate values of internal variables which can be used in workflows and documents. Nintex supports all the data types considered in this study.

An answer to each question is uniquely identified by an internal name. It can be defined as required and contain a default value. Conditional rules and formulas to manipulate variables and questions can be added to the questionnaire or workflow, and are managed centrally (see Sections 10.2.4 and 10.2.5).

The document body of a template consists of a DOCX file containing tags that are replaced with values of variables during the document automation process.

Score: 5/5

10.2.2 Template Creation

Since document automation is only one of many features of Nintex, users need to add a workflow for defining their document automation process. In our scenario, a short workflow consists of a questionnaire for user inputs and a document generation action that is triggered after the submission of the questionnaire. This process starts with adding questions to the questionnaire. Questions can be rearranged by drag-and-drop to target positions. Users can enforce conditional rules and formulas in separate tabs (see Sections 10.2.4 and 10.2.5). A preview of the current questionnaire can be displayed at all times.

The document body of the template is a regular DOCX file that is manually annotated

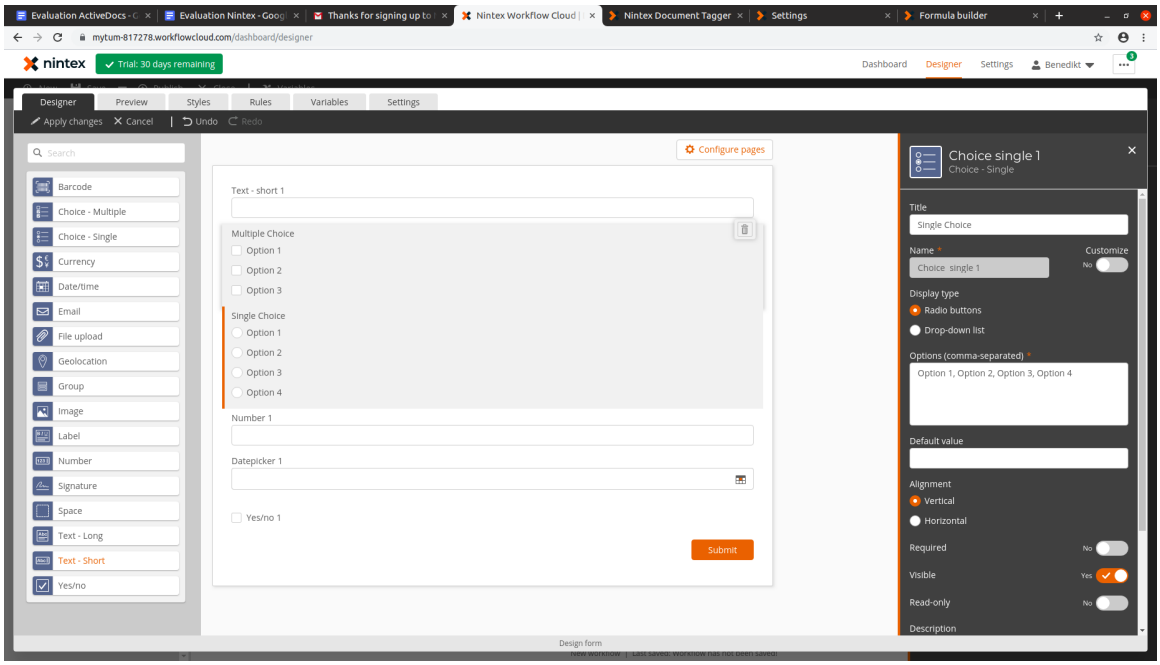

Figure 10.2: Nintex - Template Structure

with tags corresponding to Nintex variables. Users specify the location of this file inside the workflow configuration.

While the creation of questionnaires is fast and intuitive, specifying templates' locations and adding Nintex variable tags into documents can be initially challenging for non-technical users.

Score: 5/5

10.2.3 Document Generation

The document generation process in Nintex can be customized to a high degree. In the basic scenario it consists of a user answering questions from the questionnaire and one or more documents being generated after the submission. DOCX and PDF files can be generated as long as a valid file name and location are specified by the template creator.

External data sources like databases, files or data queried by API calls can be used to supplement data that has been provided by the users. This data cannot be provided to the users during the questionnaire, and is only used to alter variables after the questionnaire is submitted.

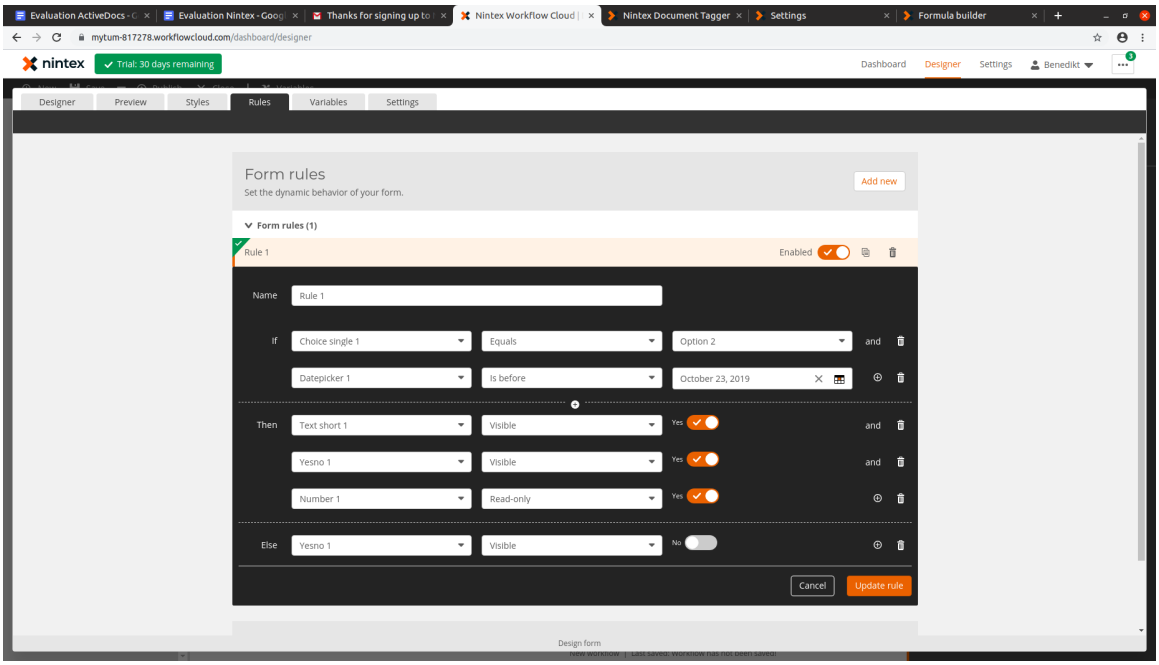

Figure 10.3: Nintex - Conditional Logic

Nintex's document automation process is flexible and intuitive but can use external data sources only to a limited degree. Nintex offers connectors to MySQL, MS SQL databases and other platforms such as Salesforce, Sharepoint, DocuSign, etc.

Score: 5/5

10.2.4 Conditional Logic

Conditional logic can be used inside a questionnaire to change variable values and to manipulate almost every part of the questionnaire. They are part of a questionnaire and managed centrally. The document body of templates cannot hold any conditional logic.

A conditional rule consists of any number of conditional expressions that compare a variable to a static value or another variable. The user can specify which variable values and questionnaire settings should change if all or at least one conditional expression is evaluated to true. Advanced conditional logic can be implemented by utilizing if-then-else constructs in formulas.

Nintex can handle most complex logical expressions with a user-friendly interface when

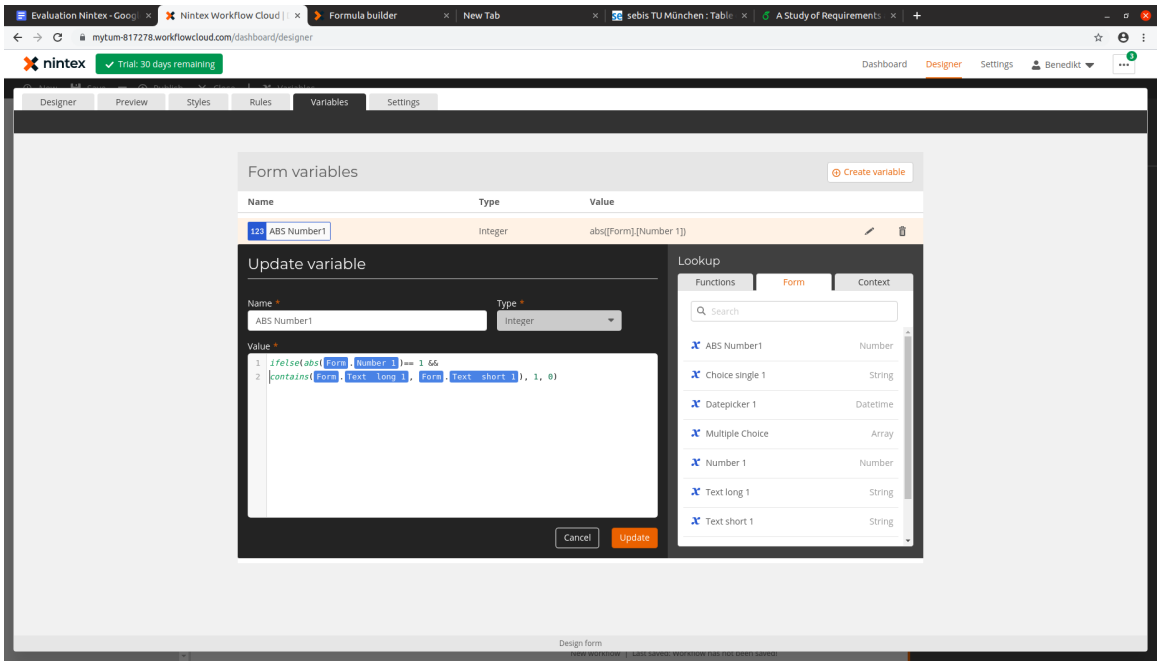

Figure 10.4: Nintex - Formulas

designing them in questionnaires.

Score: 5/5

10.2.5 Formulas

Variables can be assigned calculated values in numeric, text and date format. Formulas are managed centrally and can include any number of variables. Nintex uses basic arithmetic and exponent functions for manipulating numbers. Advanced conditional logic can be added to formulas by using if-then-else constructs.

Variables are highlighted in formulas and available functions are listed with detailed descriptions of the correct usage, return value and expected inputs. The correctness of formulas is automatically validated and locations of the syntax errors are listed at the bottom of the editor.

Formulas in Nintex are sufficient for most use cases. They can be created and validated easily with an intuitive interface.

Basic and Advanced Formulas	:	**1.7/5**
Statistical Functions	:	**1.7/5**

10.2.6 Documentation

The documentation is aimed at non-technical users and consists of a well-structured online documentation that is available to the public. All features are explained meticulously and interlinked with related articles. The documentation provides a search tool for finding Nintex Workflow features, such as designing workflows, constructing templates, generating documents, and applying conditions or formulas. Screenshots are included to illustrate expected outcomes or how features work. Instruction videos do not exist in the documentation, yet users can find them on the YouTube channel.

Score: 4.6/5

10.2.7 Usability

Creating workflows for document automation in Nintex is not intuitive for non-technical users. However, the process of designing questionnaires is straightforward and customizable. Nintex provides mechanisms to prevent, diagnose, and fix errors during the process of template creation. The user interface is aesthetic and minimalist with all important features being displayed in the main view.

In terms of logical expressions, Nintex provides two methods for users, either by defining conditions as workflow tasks with two branches for true and false results, or declaring conditional statements while designing forms. For complex conditions, this results in very big and difficult to manage decision trees.

Score 4.2/5

10.3 Evaluation of Supporting Features

10.3.1 Integration

A workflow in Nintex can be triggered by an incoming HTTP request containing variable values enabling the user to integrate document automation into larger automated processes.

10.3.2 Document Management

Nintex does not provide integrated document management functionalities. However, it supports various external cloud-based document management systems, profiting from their powerful features.

10.3.3 User Management

Administrators can create and manage users and user roles. One can assign user roles to users and synchronize the user management with Microsoft Azure Active Directory. Permissions to access generated documents have to be managed outside of Nintex. There are three relevant user roles:

- ”Administrators”: Have all permissions and can manage users
- ”Designers/Developers”: Can create templates and workflows
- ”Participant”: Can use workflows to trigger the generated documents

User management in Nintex can model most organizational structures but falls short in managing access to generated documents.

10.3.4 Collaboration

Nintex provides an option to share the templates with other users and user groups. Further collaboration capabilities have to be manually implemented into workflows.

10.3.5 Enterprise Scalability

Nintex is designed for large enterprises and can not only automate the document generation process, but also model business processes that go beyond the scope of document automation. Priority support is available to help with user problems or the setup of complex workflows.

10.3.6 Advanced Authentication

Nintex provides Single Sign-On and user management through Microsoft Azure Active Directory to create a seamless integration with the existing system environment.

10.3.7 Data Ownership

Nintex is completely cloud-based. However, it lets users decide among various data centers around the world, including Europe, Australia, and Japan, thus addressing possible concerns about data residency and specific regulatory compliance.

Document bodies of templates and documents are not stored in the cloud resulting in a higher degree of control over own data although documents cannot be generated without uploading sensitive data to the cloud.

11 Precisely

Precisely is a web-based document automation tool that only runs on a public cloud. It is developed by the company with the same name since 2014. It is focused on the creation and management of contracts and provides capabilities for E-Signing contracts.

11.1 Executive Summary

Precisely supports most of the input types required in document automation, except for multiple selectors. The tool provides a validation mechanism to prevent entering erroneous data into the template. Conditional expressions can be applied to help users logically customize the content of their templates, yet Precisely does not support combined logical expressions with AND/OR operators. Additionally, mathematical operators cannot be applied to logical expressions.

To compose templates in Precisely, users create questionnaires to define input data and allocate the corresponding answers to the desired position(s) in the templates. After that, the template needs to be published for the end users to fill it in. Finally, the data from the filled questionnaire(s) are used to generate DOCX or PDF files.

In terms of usability, the user interface of Precisely is intuitive, elegant and minimalist. All important features can be accessed from the main interface. Different components in the template are highlighted to help users visualize the static and dynamic elements.

Precisely offers a function to create user groups and definition of access rights for collaboration among involved parties. Additionally, the tool provides APIs for document and template management.

Precisely is most useful to generate documents that do not require complex conditional logic or mathematical formulas. The tool's potential is fully utilized when its E-Signing capabilities are integrated into the document automation process and documents are generated in a collaborative environment.

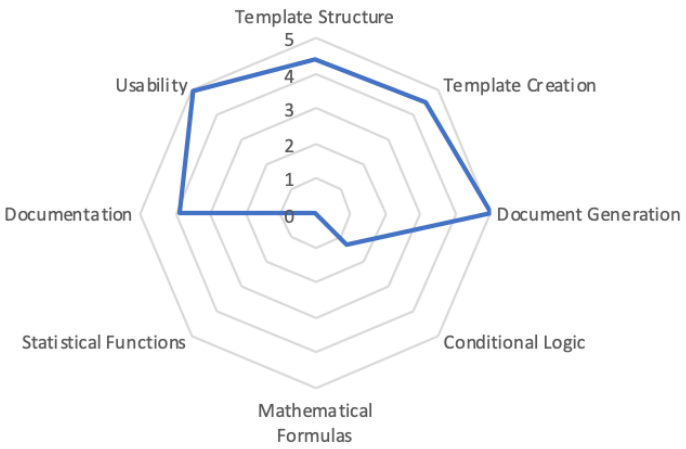

Figure 11.1: Radar diagram for Precisely - Core Document Automation Functionality

Integration	✓
Version Control	-
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	✓
○ Single Sign-On	-
Data Ownership	
○ Cloud-based	✓
○ On-premise	-

Table 11.1: Supporting features of Precisely

11.2 Evaluation of Core Document Automation Functionality

11.2.1 Template Structure

A template in Precisely consists of a questionnaire and one or more documents that are generated at the same time. A questionnaire consists of any number of questions split among multiple pages. Each question is uniquely identified by a reference name and can

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	-
Table	✓

Table 11.2: Template structure features of Precisely

contain a default value. Precisely supports most of the required input data types, except for multiple selector.

A document inside a Precisely template consists of static text containing conditional blocks (see Section 11.2.4) and references to questions. Formatting like paragraphs, page breaks and clause lists can be applied to the template.

Templates in Precisely are clearly separated into documents and user inputs making them conveniently reusable in other templates.

Score: 4.4/5

11.2.2 Template Creation

In Precisely's web-based interface, templates can be created from scratch, or imported from a DOCX file or existing Precisely templates. When a DOCX file is imported, some formatting might be lost. Users can write static text into the document, and add references to questions by dragging and dropping questions into the template. These references are replaced with answers during the document generation process. Additionally, conditional text blocks can be created by selecting a text, marking it as conditional and adding a conditional expression (see Section 11.2.4). They can be converted back to static text with one click.

Questions and documents can be copied and reused in other templates. Changes to a document inside a template are saved every few seconds and are immediately effective for new documents that are generated. The date of the last modification of a template is displayed but older versions are not saved.

The templates of Precisely are created flexibly through a combination of text editor and

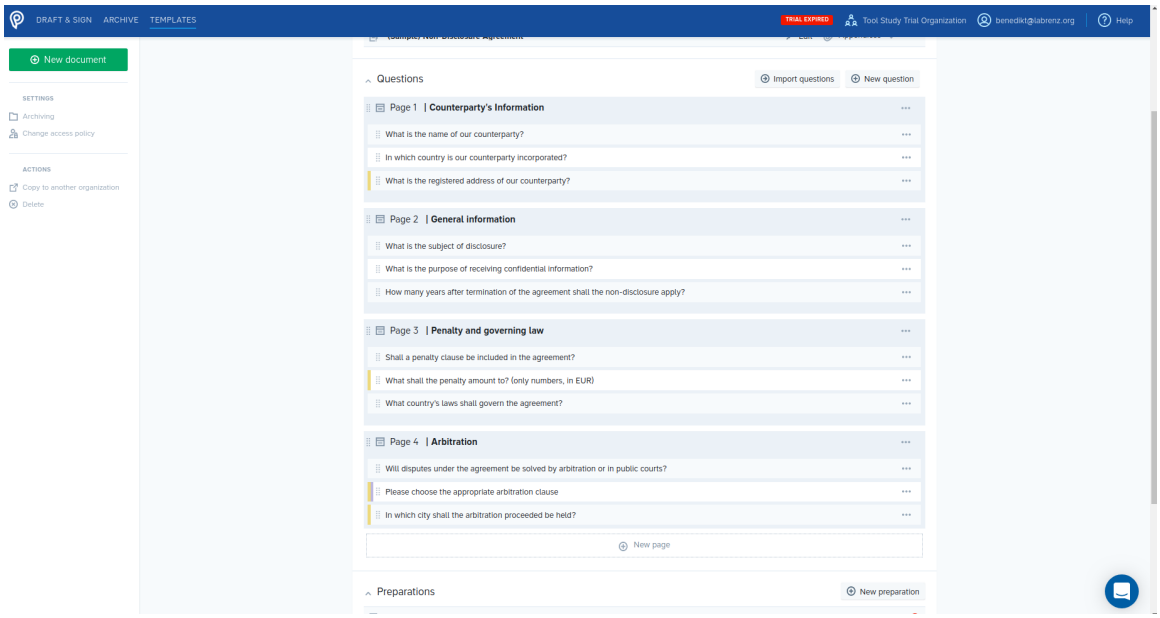

Figure 11.2: Precisely - Questionnaire Creation

drag-and-drop but lack a version control system to access the old version.

Score: 4.5/5

11.2.3 Document Generation

If users define a checklist of preparation tasks for document automation, they have to tick off all items of the list before proceeding. Users can choose a location within Precisely's document management system or use a default location. Then they answer all questions from the template, which can be split over multiple pages. Questions can be annotated with explaining help texts and all answers can be reviewed before submission. User inputs cannot be substituted with a database or a file upload.

After documents of a template have been generated, users can make changes in a text editor and download them as DOCX and PDF file. Signees for E-Signing can be added and the E-Signing process be triggered. Additional PDF files can be uploaded as appendices.

Precisely provides a smooth document generation process that includes E-Signing but is not able to use databases as data sources.

Score: 5/5

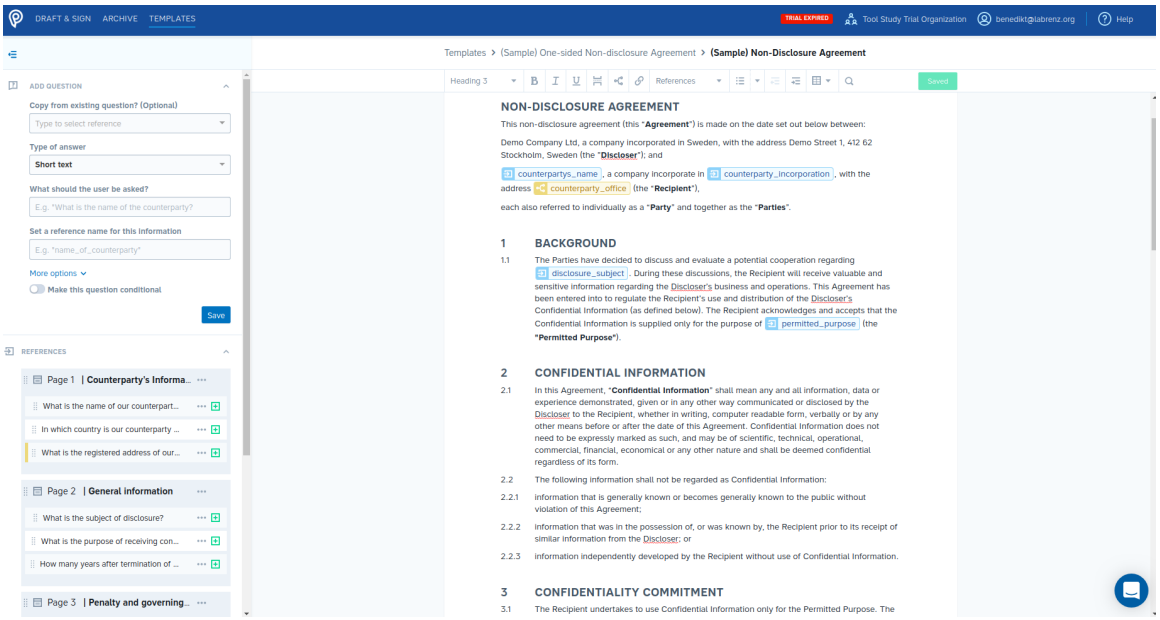

Figure 11.3: Precisely - Document Body Creation

11.2.4 Conditional Logic

Conditional logic can be used to decide if a question or text block is displayed. A conditional expression in Precisely consists of a comparison of any question value with a static value. Precisely does not support compound conditional expressions, only a single comparison is allowed. The available operators depend on the question's data type and can be "is equal", "is not equal", "less than", "greater than", "less than or equal", "greater than or equal", "is any of". The static value in the comparison also depends on the questions's data type to prevent invalid expressions.

Lack of the capability to chain conditional expressions does not allow questions and text blocks to depend on multiple questions which is important in many use cases.

Score: 1.3/5

11.2.5 Formulas

Currently, Precisely does not provide support for mathematical calculations.

Basic and Advanced Formulas	:	**0/5**
Statistical Functions	:	**0/5**

11.2.6 Documentation

The documentation is aimed at non-technical users and is available online for the public. It consists of a number of articles that describe how the main features work, but does not describe how to generate PDF or DOCX documents from templates. Precisely illustrates feature usages with explanatory screenshots, making them easily understandable to readers. However, there are no instruction videos in the documentation.

Score: 3.9/5

11.2.7 Usability

Precisely is a web-based tool which requires minimal effort to create questionnaires, draft templates and generate documents. Its user interface is intuitive and minimalist. Precisely presents all important features directly on the main display and uses commonly known graphical elements to illustrate the features of the system. The process of creating questionnaires for templates is very flexible. Users can create questionnaires from the template dashboard or inside the template editor. Moreover, in the process of drafting templates, users can undo and redo actions. Precisely also prevents users from making errors through input validation and provides meaningful messages to explain the source of arisen errors. The documentation helps users to understand the software features, but lacks video tutorials.

Score 4.9/5

11.3 Evaluation of Supporting Features

11.3.1 Integration

Precisely provides a REST API with endpoints for all important functionalities like managing templates, generating documents from templates and the integrated E-Signing process.

11.3.2 Document Management

After a document has been generated from a template, it will be saved in a location within the cloud that has been specified at the beginning of the document automation process. A list of all documents that a user has access to is available and can be filtered by owner, category and current status. The available statuses are ”Draft”, ”Pending”, ”Signed” and ”Imported”, which describes documents that have been uploaded but are not generated by Precisely. Additionally, users can search for documents generated by Precisely using full-text search. Metadata of documents is sparse and limited to document name, date of last modification, owner and current status.

Except for version control, Precisely offers core functionalities that are expected of a document management system but tracks only a limited amount of metadata.

11.3.3 User Management

Permissions in Precisely are based on user roles. Additionally, user groups can be used to manage access to templates and documents. There are three different user roles in Precisely:

- ”Admin”: Can invite and manage users. Has full access to all templates and documents.
- ”Manager”: Can create and edit templates. Can generate documents from templates and see all other documents that have been generated from the templates that are shared with the user. Is allowed to make changes to the text of documents after they have been generated.
- ”Member”: Can generate documents from templates that are shared with the user but cannot make changes to the text afterwards. Can only see documents generated in their user group.

User roles in Precisely map to common organizational roles and can cover all required use cases.

11.3.4 Collaboration

Users can share templates and documents with user groups or with the whole organization with the level of access based on the respective user roles of team members. Generated documents can be commented on by all users that have access and the documents’ content can be changed collaboratively if users have sufficient permissions.

Precisely enables users to ask managers for approval of draft versions of generated documents and send them for signing if they are ready to be signed.

Precisely offers many functionalities for a collaborative document automation process from end-to-end.

11.3.5 Enterprise Scalability

Enterprise customers can contact technical experts directly in addition to the regular customer service.

11.3.6 Advanced Authentication

Precisely supports Two-Factor Authentication using the Google Authenticator mobile app for enhanced security.

11.3.7 Data Ownership

Precisely is a cloud-based tool that cannot be installed on premises and cannot run locally in a web-browser without uploading user inputs to the cloud. Users have limited control of the document generation service, templates and their data compared to an on-premise installation.

12 SmartDocuments

SmartDocuments is a cloud-based document automation tool with a template editor that is installed on a local machine and linked to the cloud. Documents can also be generated from an app that is available for iOS and Android. The tool is developed by the company with the same name since 1994.

12.1 Executive Summary

SmartDocuments requires interactions with both web application and client-side software for creating templates from questionnaires and/or data sources. The tool supports all of the input types considered in this study. SmartDocuments supports compound logical expressions and basic arithmetic operations during the template creation process.

To generate documents from templates in SmartDocuments, users create a template on a web interface first. Next, the template is edited by a software installed locally on the user's computer. Dynamic elements in templates are constructed as input fields or retrieved from data sources. After saving the templates, users answer the templates' questionnaires in another webpage to provide data for dynamic input fields. Finally, SmartDocuments generates documents in different file formats for users to download them.

Composing a template in SmartDocuments with input parameters is intuitive, but using data sources is challenging. Additionally, the client-side editor is prone to error. If users perform unexpected interactions with the system, it can lead to a crash which may erase all the static elements of the template. The tool also does not provide documentation or help in its editor or web page interfaces.

SmartDocuments offers a version control for its templates. The tool has user management features to provide collaboration for various parties involved in the preparation of templates.

Overall, SmartDocuments helps users generate documents from various sources, but the process may be inconvenient.

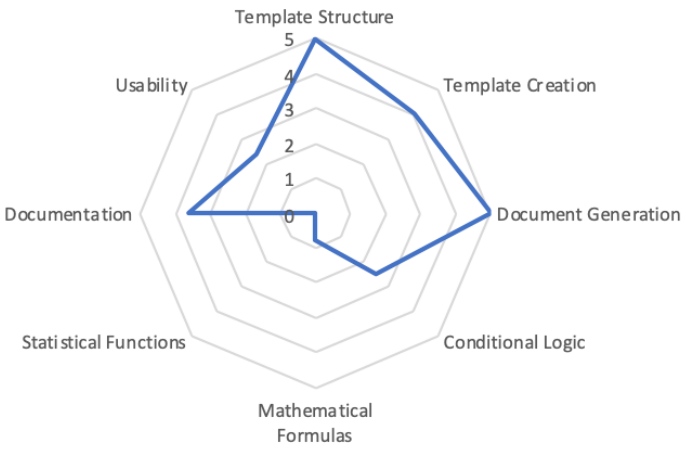

Figure 12.1: Radar diagram for SmartDocuments- Core Document Automation Functionality

Integration	✓
Version Control	✓
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	-
○ Single Sign-On	✓
Data Ownership	
○ Cloud-based	✓
○ On-premise	✓

Table 12.1: Supporting features of SmartDocuments

12.2 Evaluation of Core Document Automation Functionality

12.2.1 Template Structure

Templates in SmartDocuments consist of a questionnaire and a document body. The questionnaire has a tree structure that determines the data type of answers, order of questions

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 12.2: Template structure features of SmartDocuments

and follow-up questions if specific answers to previous questions are selected. Questions are grouped in pages and can be annotated with explaining help texts, marked as required for user input and contain a default value. SmartDocuments supports all the data types required in this study.

The document body consists of formatted, static text in which references to answers of questions can be inserted. For questions of the types ”Select”, ”Multi-Select” and ”Yes/No”, users can either select answers from an option list, or insert a new text.

The questionnaire tree helps users to keep a clear overview of complex structures, as well as to keep track of the links between answers and their usages in the document.

Score: 5/5

12.2.2 Template Creation

Templates are initially created in the web interface of SmartDocuments, then they are checked out and downloaded to a standalone editor installed on the user’s machine. New templates can be created from scratch or existing ones in the user’s repository. The layout for documents generated from the template can be defined centrally and applied to each template. Questionnaires are created by adding pages containing questions and answers to the question tree. Templates contain static text written by users in which they can insert references to answers by drag-and-drop.

Text blocks with the associated questions from the questionnaire can be stored and inserted into other templates. They are centrally managed in the cloud and if a text block is updated, the change is automatically propagated to all templates in which it is used.

SmartDocuments provides an elaborated version management distinguishing between a

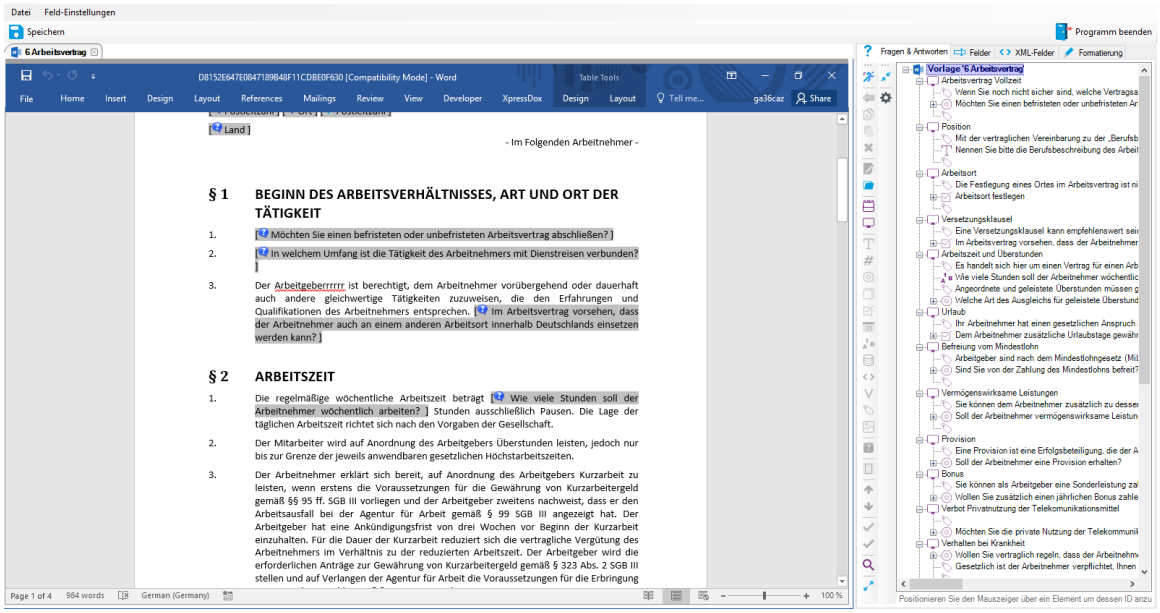

Figure 12.2: SmartDocuments- Template Structure

production version and a development version for each template. To make changes to a template, it has to be checked out first in order to prevent other users from working on a conflicting version. After the changes have been saved in the editor, the updated template is automatically uploaded to the cloud and a new development version is created. The latest development version can be published from the web interface to create a new production version.

Template creation in SmartDocuments requires little effort as components of templates can be modularized.

Score: 4/5

12.2.3 Document Generation

The document automation process in SmartDocuments is triggered from a separate web interface that is available to end users. The user is presented with a questionnaire which can be split over multiple pages. A preview of the generated document can be displayed next to the questionnaire with the questions and their corresponding sections in the documents being highlighted. User inputs can not be substituted by external data sources but can be complemented by utilizing the question type ”database” during the template creation.

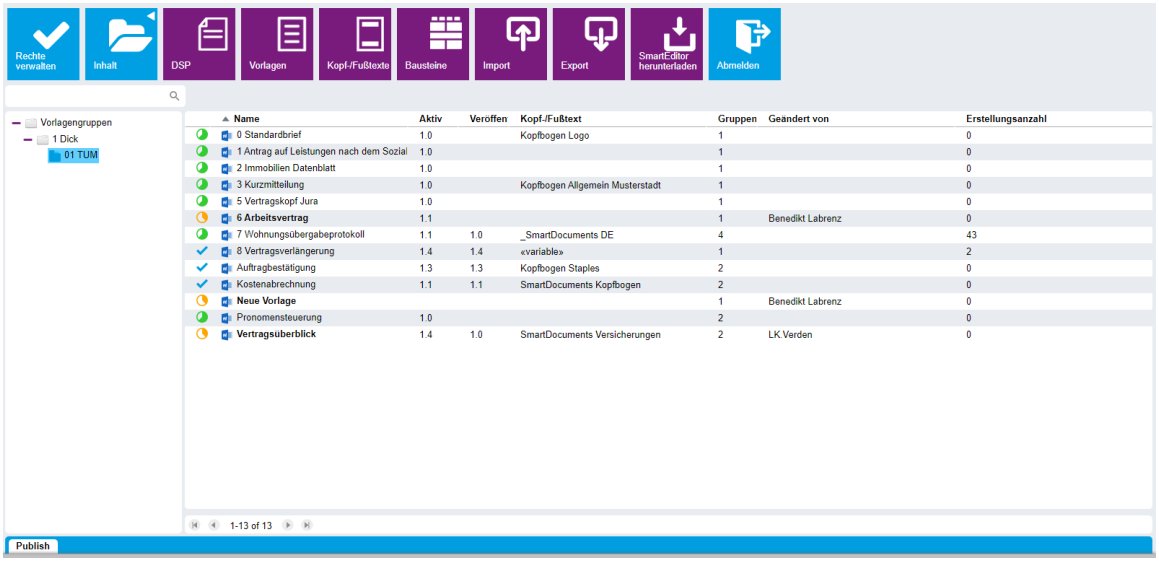

Figure 12.3: SmartDocuments- Template Management

After all questions are answered, users are presented with an overview of all answers and can jump back to any location in the questionnaire to make changes. After the answers are submitted, the document is generated and automatically downloaded as a DOCX file.

The preview during the document generation in SmartDocuments helps with selecting the right answers but can take a while to be loaded. SmartDocuments can import XML data sources and query data from MS SQL, MySQL, Oracle, and PostgreSQL databases.

Score: 5/5

12.2.4 Conditional Logic

Conditional logic in SmartDocuments is modeled in the questionnaire tree by utilizing Word's Field Codes and the question types "Select", "Multi-Select" and "Yes/No". Answers to these questions can be used to insert alternative versions of words, phrases or text blocks into the document. Follow-up questions are only displayed if a specific answer to a previous question of these types was selected.

While offering a clear overview of dependencies inside the questionnaire tree, the implementation of conditional logic in SmartDocuments is limited within the capabilities of Word's Field Codes. Other question types like numeric or text user input can be used in conditional expressions, but users need to familiarize themselves with the syntax of defining conditional statements in Word, which non-technical users may find challenging. If

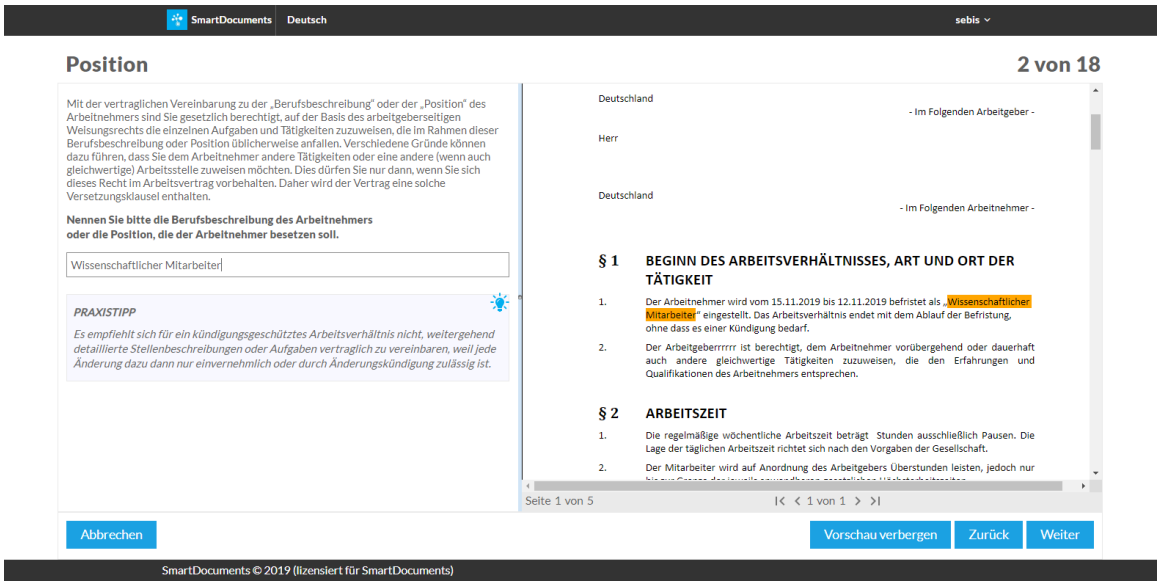

Figure 12.4: SmartDocuments- Document Generation

multiple answers can lead to the same follow-up questions, then these questions have to be inserted multiple times.

Score: 2.5/5

12.2.5 Formulas

Similar to the definition of conditional logic, SmartDocuments supports basic arithmetic operations inside templates using Word's Field Codes. To construct arithmetic expressions, users first need to create a question and then drag it into a desired position in Word's page. Next, right-clicking on the dragged element and selecting "Toggle Field Codes" will reveal its Word's Field Code syntax. Finally, users drag the target elements into the expression and place an arithmetic operator between them. For example, to construct an $x + y$ expression where x and y are number questions, the syntax is $\{= x + y\}$. Note that x and y need to be dragged from the question tree into the Word Field Code.

Basic and Advanced Formulas	:	**0.8/5**
Statistical Functions	:	**0/5**

12.2.6 Documentation

The documentation for SmartDocuments consists of structured files available in multiple languages that are supplied by the customer service. They describe the features available

from the web interface and the editor used for the template creation, and the template creation process as a whole. Features are described in detail and the documentation is aimed at non-technical users. Some features like the question type "database" are missing in the documentation. Searching is done via the search tool of document reader applications, hence users can only search for terms syntactically, i.e. only texts matched with search terms are displayed.

In conclusion, documentation in SmartDocuments is concise and well-structured with explanatory screenshots provided as examples. Searching for key words in documentation is quite limited because it relies on a syntactic search mechanism of the user's document reader application.

Score: 3.6/5

12.2.7 Usability

There are noticeable drawbacks of SmartDocuments regarding user experience. First, users must work with three separate components in the document automation process: SmartControl - a web application to manage templates; SmartEditor - a SmartDocuments client-side template editor; and Wizard - a webpage to answer questionnaires of the template. In order to load the created templates, users must use client-side software, which requires an Internet connection to the cloud server of SmartDocuments. This complicates the process of document automation, making it cumbersome and inconvenient. Furthermore, when users save the template after interacting with certain input types, the client-side template editor can behave abnormally and crash. Additionally, there is no accessible documentation or help tool in the template editor, so users cannot search for instructions of using particular system features or explore the capabilities of SmartDocuments.

On the other hand, SmartDocuments has a good validation mechanism, e.g. to prevent users from inputting alphabet characters into numeric or date fields. Additionally, the tool maintains a consistent design and language across different user interfaces.

Score 2.4/5

12.3 Evaluation of Supporting Features

12.3.1 Integration

SmartDocuments provides a REST/SOAP API to generate documents from templates.

12.3.2 Document Management

SmartDocuments does not offer a document management system.

12.3.3 User Management

Administrators can create, manage and delete user accounts. Users can be added to permission groups that manage the access to a set of templates.

Although the complexity of SmartDocuments ' user management is kept low, it can handle the requirements for most use cases.

12.3.4 Collaboration

Templates can be created in a team of multiple users each creating a new version with a team lead publishing the changes after review. Advanced features for collaboration or communication are not provided.

12.3.5 Enterprise Scalability

The high degree of modularization and the powerful version management in SmartDocuments makes it easy to scale for large organizations. Documents can be generated from the web interface, integrated software tools and apps on mobile devices. Technical support is available to configure SmartDocuments and to help with the template creation.

12.3.6 Advanced Authentication

SmartDocuments supports Single Sign-On for the system of end users, so that they can login to SmartDocuments using the user's system credential, fill in questionnaires and generate documents. However, SmartDocuments does not provide SSO to the SmartControl system where templates are managed.

12.3.7 Data Ownership

SmartDocuments can be installed on premises or hosted in the cloud giving the users a lot of control over their data. Templates can be downloaded and shared offline between different instances of the tool.

13 Templafy

Templafy is a cloud-based document automation tool that can also be run offline in a Microsoft Word Add-In. It is developed by the company with the same name since 2014 that started out with the goal to create documents that adhere to corporate design.

13.1 Executive Summary

Templafy supports most of the input types considered in this study, except for customizable numeric values. Input validations, masking, and advanced mathematical or conditional expressions are defined via Visual Basics for Applications (VBA) helpers in Templafy. By default, Templafy supports declaration of a single conditional expression.

To generate documents in Templafy, template managers create a document containing static and dynamic text elements in the Templafy repository and then save this template. End users can then access this template in their Templafy account. Next, users create profiles and provide values to dynamic text elements by answering the template's questionnaire. Finally, based on the answers, Templafy generates a Word document which can be modified or exported as a PDF file.

In terms of usability, Templafy is quite intuitive to non-technical users with minimalist, consistent and aesthetic design. However, by default, Templafy does not validate users' answers so inputs with unexpected syntax are accepted by the system. Additionally, access to documentation is not straightforward from the Templafy interface. From the Word's *Home* toolbar, users need to click on Templafy $>$ About $>$ Get Support to reach Templafy's online documentation.

API endpoints are available for external systems to use document automation features of Templafy. The tool provides features to manage user groups and access rights, Single Sign-On and Two-Factor Authentication. Templafy leverages collaboration features provided by Microsoft Word.

In general, Templafy provides a user-friendly interface for automating documents. Nearly all elements of templates in Templafy are reusable and managed centrally. Templafy does not have built-in version control but combining the tool with external document management systems can provide this feature to users.

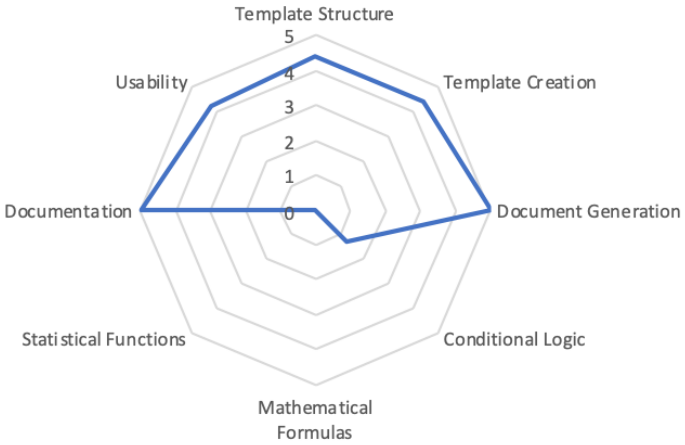

Figure 13.1: Radar diagram for Templafy - Core Document Automation Functionality

Integration	✓
Version Control	-
User Management	-
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	✓
◇ via Mobile App	✓
○ Single Sign-On	✓
Data Ownership	
○ Cloud-based	✓
○ On-premise	-

Table 13.1: Supporting features of Templafy

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	-
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	✓
Table	✓

Table 13.2: Template structure features of Templafy

13.2 Evaluation of Core Document Automation Functionality

13.2.1 Template Structure

Templates in Templafy consist of a questionnaire and a document body. A questionnaire comprises any number of questions that are split over multiple reusable questionnaire elements. Available data types for questions are listed in table 13.2:

Similar to the questionnaire the document body is also modularized and consists of any number of reusable text elements. Text elements consist of static text annotated with links to questions from a questionnaire. They can be partially or completely shown or hidden based on user input (see Section 13.2.4). An underlying base template is used to apply uniform formatting to the template.

Templates in Templafy are highly modularized which makes them easily reusable. However, the tool does not provide a dedicated question type for numeric values. **Score: 4.4/5**

13.2.2 Template Creation

Templates are created and managed in a web-based interface for administrators. Reusable elements for the questionnaire and the document body can be added, rearranged and deleted. These elements are centrally managed in the same user interface.

Text elements are created and updated with a Microsoft Word Add-In. Templates can be created from scratch or existing DOCX files. Users write static texts into the document and add links to questions from a questionnaire element. Text blocks can be made conditional based on the value of a Templafy element.

After changes have been made to a questionnaire element or text element it has to be saved and uploaded to Templafy. The changes are immediately available for all documents that

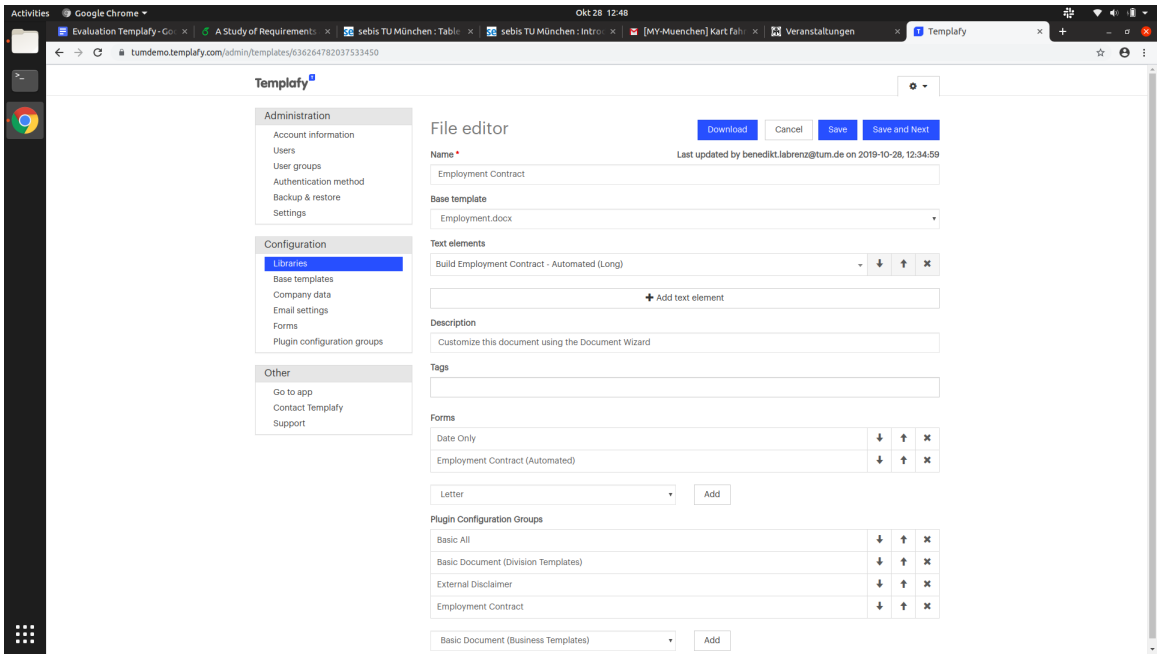

Figure 13.2: Templafy - Template Configuration

are generated from the modified template.

Templafy's modularized components make templates easily reusable. Yet for new users, tracing the connections of a dynamic element in a document body to their corresponding questionnaire can be confusing.

Score: 4.4/5

13.2.3 Document Generation

The document automation process in Templafy is usually started from a web interface, where template managers create new templates. Next, Microsoft Word Templafy Add-In is used to insert static and dynamic text elements into templates. This add-in can also generate documents in offline mode. When end users access the templates in Word, they are presented with a list of questions. After all questions that are marked as required are answered, a document is generated as a DOCX file. If templates are customized with Templafy VBA helper, more complex deployment scenarios are possible.

Templafy's document automation process is straightforward and does not offer many advanced features by default. These features can be customized with Templafy VBA helper yet certain programming knowledge in Visual Basic is required. Besides DOCX docu-

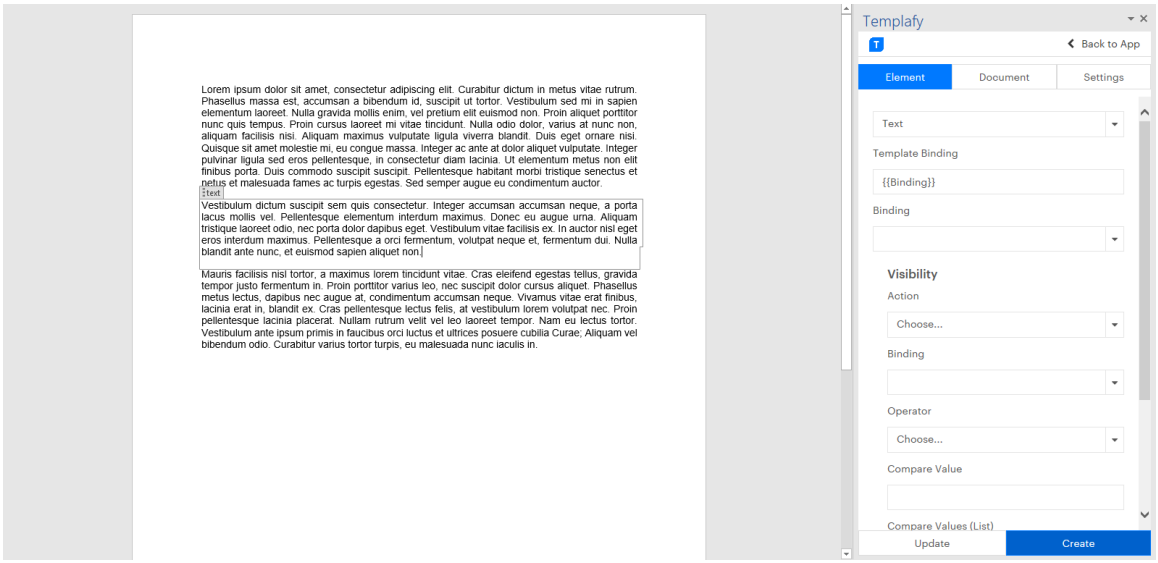

Figure 13.3: Templafy - Template Structure

ments, the tool can also generate spreadsheets and presentation slides.

Score: 5/5

13.2.4 Conditional Logic

Text blocks and words or phrases can be displayed conditionally while questions can not be made conditional.

A conditional expression contains a reference to an answer to a question in a template questionnaire, a logical operator, and a comparable value or list of constants. The result of conditional expressions is used to decide if a text element is displayed or hidden in the document. The available operators are ”equals to” and ”does not equal to” for a single static value, and ”in” and ”not in” for lists of static values.

By default, conditional expressions in Templafy are limited to basic comparisons and cannot be chained together to model more complex business logic.

Score: 1.3/5

13.2.5 Formulas

Currently, Templafy does not support integrated formulas by default.

Basic and Advanced Formulas	:	**0/5**
Statistical Functions	:	**0/5**

13.2.6 Documentation

The documentation is searchable and split into multiple sections to address different audiences from non-technical users to developers. Each section contains a variety of detailed articles that explain key features from create, edit and reuse templates to short tutorials. However, basic usage of conditional expressions is not included in Templafy's documentation, only advanced conditional logic configuration in JSON and Form are mentioned.

Score: 4.7/5

13.2.7 Usability

Templafy document automation process is considerably user-friendly in general, creating templates and text elements with the Microsoft Word Add-In is fairly intuitive.

Conditional logic is implemented by setting bookmarks following a rather technical naming convention. Reusing templates in Templafy is flexible and straightforward.

On the other hand, the tool can improve its validation mechanism to prevent faulty inputs from entering the document automation process. For example, date input fields shall not accept alphabetic characters. Additionally, documentation should be placed in a more visible position in Templafy's Add-in.

Templafy's user interface is consistent, intuitive, and minimalist with all relevant features being easily accessible within the same interface.

Score 4.2/5

13.3 Evaluation of Supporting Features

13.3.1 Integration

Templafy provides a RESTful API with endpoints for all important functionalities like managing or updating templates or text elements. Generating documents by using the API is currently not supported. Templafy can be integrated with a variety of other tools to

trigger the document automation process and to add advanced deployment of the generated documents.

13.3.2 Document Management

Templafy does not offer a document management system but can be configured to use an existing one.

13.3.3 User Management

There are three different user roles in Templafy, each including all permissions of lower rank user roles:

- ”Super Administrator”: has access to all configurations like authentication or backups.
- ”Administrator”: can create, manage and delete templates and reusable elements for templates. Can manage users and user groups.
- ”User”: can generate documents from templates.

User groups can currently only be used to give a limited group of users access. The user roles in Templafy can model most of the organizational roles but user groups can only be utilized in a very limited way.

13.3.4 Collaboration

As a Word Add-In, Templafy leverages the comment, chat, and share documents features from Microsoft.

13.3.5 Enterprise Scalability

The high degree of reusability in Templafy makes it easy to scale for large organizations. The tool supports Single Sign-On, Microsoft Active Directory and many pre-built connectors, important for its integration into an existing IT landscape.

13.3.6 Advanced Authentication

Templafy offers Two-Factor Authentication via Email. Additionally, Microsoft Authenticator App can also be used to protect documents stored on Microsoft’s cloud-based file management system.

13.3.7 Data Ownership

Templafy is primarily cloud-based but can also be run in offline mode using the Microsoft Word Add-In. In this case no sensitive data is uploaded to the cloud during the document automation process. Templates and reusable elements are stored in the cloud but can all be uploaded to another file system.

14 Windward

Windward is a document automation tool that is only installed on premises. Templates are created in a Microsoft Word Add-In and can be deployed to a server and made available through a REST API or integrated into custom software. The tool is developed by Windward Studios since 2003.

14.1 Executive Summary

Windward is a powerful tool to automate documents from data sources. Except for multiple selectors, Windward supports all other input elements required in this study, as well as the capability to query and filter results from data sources. Users can construct logical expressions at Level 4 complexity (see Section 2.1.4), along with basic and advanced mathematical operations, except for logarithm.

The tool is able to generate documents with dynamic content from data sources or input parameters. Users configure the locations of these dynamic elements in their templates. If input parameters are used, Windward prompts for their value before generating DOCX documents.

Windward's user interface may not be intuitive for non-technical users. Using Windward features requires certain technical knowledge to understand the required syntax or commands. The tool offers validation mechanisms to prevent invalid data from being accepted. Documentation is well-structured and illustrates procedures of using the tool's features with screenshots and simple language.

As a Word plugin, collaboration, Two-Factor Authentication, user and access right management features are offered by Microsoft. Windward enables integration of document automation into a server with its RESTful, Java and C# report engines.

To summarize, Windward is a designated tool to generate documents from data sources. Technical users may quickly adapt to the user interfaces and features of the tool, but non-technical users may find it challenging to familiarize themselves with the system.

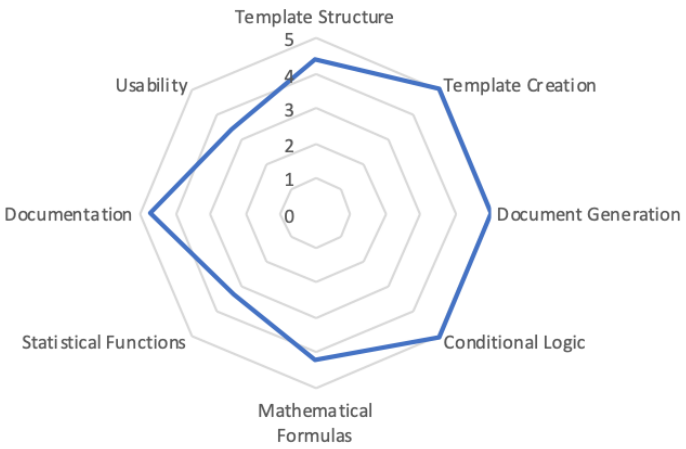

Figure 14.1: Radar diagram for Windward- Core Document Automation Functionality

Integration	✓
Version Control	-
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	✓
○ Single Sign-On	✓
Data Ownership	
○ Cloud-based	-
○ On-premise	✓

Table 14.1: Supporting features of Windward

14.2 Evaluation of Core Document Automation Functionality

14.2.1 Template Structure

Templates in Windward are made up of a list of user inputs and a document body annotated with tags. User inputs can be marked as required and can have a default value but can not be displayed conditionally. Windward supports all the data types considered in

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	-
Table	✓

Table 14.2: Template structure features of Windward

this study, except for multiple selectors.

The document body consist of formatted, static text and tags containing values from user inputs or data sources that can be manipulated by using conditional logic (see Section 14.2.4) and formulas (see Section 14.2.5). Other tags can be used to repeat dynamic text patterns or to insert charts.

Score: 4.4/5

14.2.2 Template Creation

Templates are created in a Microsoft Word Add-In and can then be deployed to a server or integrated into a Java or .NET application. A new template can be created from scratch, existing DOCX documents, or from various data sources.

A list of user inputs can be created which can be used in the document body or as parameters for queries to external data sources. Users can fill the document body with formatted text and insert tags acting as placeholders for values of user inputs. Additionally, users can enforce logical conditions to display text blocks.

User inputs such as definitions, text blocks and tags can be stored in a proprietary file format in order to be reused in other templates. Changes made to such data are not automatically updated in other templates where template parts are used. After applying a modification, a template can be tested with arbitrary user input values. The code necessary to integrate the template into users' software projects can be automatically generated. Templates and files for reusable template parts are not managed by Windward and have to be shared manually within a team before they are deployed to the server.

The template creation process in Windward is flexible but requires certain technical knowl-

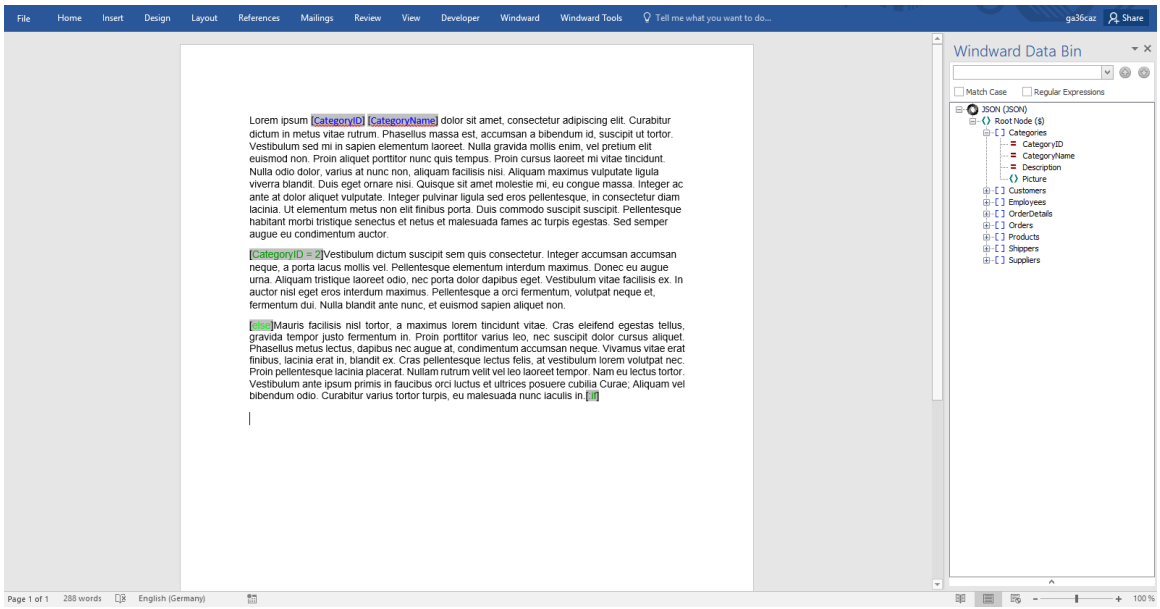

Figure 14.2: Windward- Template Structure

edge from users.

Score: 5/5

14.2.3 Document Generation

The manual document generation process in Windward starts from the Add-In by opening a selected template. Users can select the desired file format for the generated document and are presented with a list of required inputs. Supported file formats include DOCX, PDF and HTML.

Templates can be configured to complement or substitute user inputs with external data sources like databases or JSON files. After answering all questions, the document is generated and opened in a new window of Microsoft Word where it can be edited further and saved.

The document generation process in Windward is fast but may not be intuitive for non-technical users. Windward supports a variety of data sources such as XML, JSON, MySQL, PostgreSQL, IBM DB2, SQL Server, Oracle, OData, Salesforce, and Sharepoint.

Score: 5/5

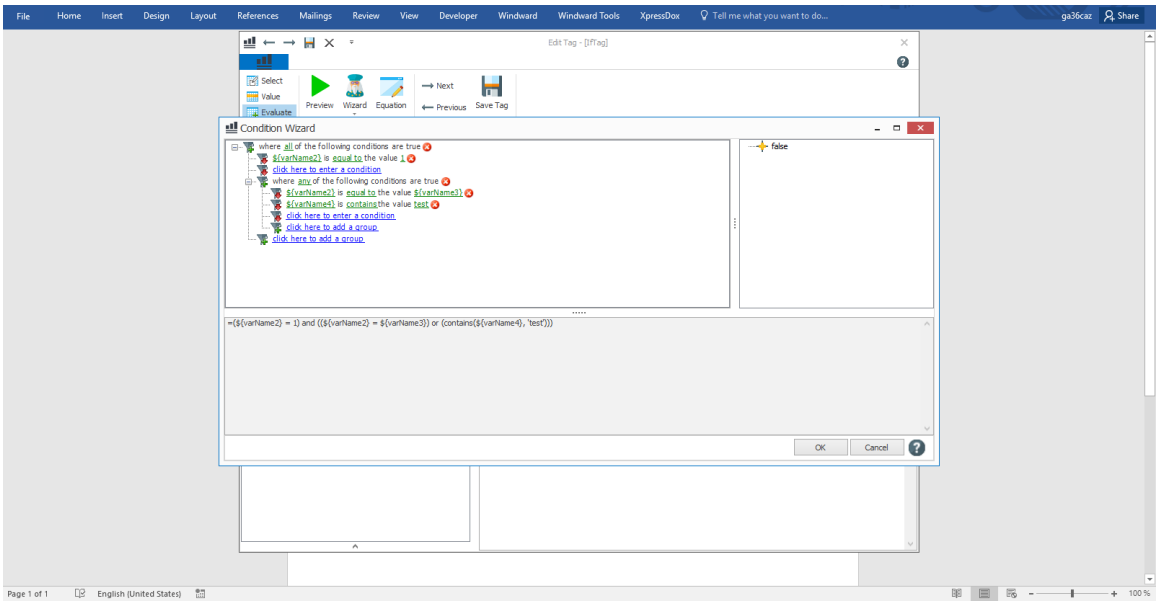

Figure 14.3: Windward- Conditional Logic

14.2.4 Conditional Logic

Conditional expressions can be attached to words, phrases and text blocks to decide whether they are displayed or hidden, or if an alternative text should be displayed. A logical expression in Windward consists of any number of comparisons of user input values with static values or other user input values. The comparisons can be nested using any combination of brackets and chained together using ”AND” or ”OR” operators. The available operators for comparisons are ”equal to”, ”not equal”, ”greater than”, ”greater than or equal”, ”less than”, ”less than or equal”, ”contains” and ”starts with”.

Conditional expressions can be built with a wizard to avoid syntax errors by constructing and displaying complex expressions using natural language and tree structures. Conditional expressions can be tested during the template creation and, in case of error, an error message with its stack trace is displayed that helps to locate the error within the conditional expression. To model more complex business logic, formulas can be included in conditional expressions (see Section 14.2.5).

Building complex conditional expressions with the wizard is intuitive and ensures that no syntax errors are made.

Score: 5/5

14.2.5 Formulas

Formulas in Windward can be applied to conditional expressions or values in tags to insert calculations based on user inputs into the document body. Available functions include basic arithmetic expressions, exponent, ceiling, floor, square root, string manipulation, date functions and regular expressions. A list of all functions with descriptions and expected input parameters is available from within the tool. Formulas can be tested and analyzed in case of syntax errors.

Formulas in Windward can be used flexibly and are easy to debug.

Basic and Advanced Formulas	:	**4.2/5**
Statistical Functions	:	**3.3/5**

14.2.6 Documentation

The searchable documentation consists of a number of tutorials that demonstrate the basic features of the tool and a large list of articles that describe advanced features in detail. Windward also provides videos and explanatory screenshots to demonstrate how features can be used.

Score: 4.7/5

14.2.7 Usability

Windward's user interface may not be intuitive to non-technical users at the first sight. Graphical elements and terms in Windward's interface are closely familiar to users who worked with databases, spreadsheets or have experience in software development. The tool offers a flexible capability to display data in tabular format from various data source structures such as XML, JSON or databases. However, because a data source is also required to construct a single selector, it is not trivial and convenient for non-technical users to use this component. Instead of inputting options into the list of the selector, users have to write their option list in XML, JSON or store it in a database.

In terms of error prevention, Windward implements user-friendly and effective mechanism to reduce the risks of accepting erroneous data. For instance, the tool provides a wizard for users to group logical conditions using AND/OR operators, select a variable name, comparison operator and compared value in logical expressions.

Score 3.4/5

14.3 Evaluation of Supporting Features

14.3.1 Integration

Templates can be deployed to a server providing a REST API with endpoints to get details on the structure of templates and to generate documents from templates. Additionally, Windward can be integrated into own Java or .NET projects.

14.3.2 Document Management

Windward does not offer a document management system. However, users can leverage version control features from external cloud-based file management systems.

14.3.3 User Management

Windward does not provide user management. Nevertheless, as a Word Add-in, users can share files with others and restrict their read or write access.

14.3.4 Collaboration

Collaboration during the creation of templates or in the document automation process has to happen outside of Windward. The tool leverages communication and sharing features provided by Microsoft.

14.3.5 Enterprise Scalability

The tool can be integrated into any software development project to automate the document generation process. In addition to regular customer service, enterprise customers can expect support with regards to the integration process and template creation. If Windward should be used without an automated process by integration with existing systems, scaling to many users becomes a very manual and chaotic task.

14.3.6 Advanced Authentication

As Word Add-in, Two-Factor Authentication and Single Sign-On are possible in Windward if users have a Microsoft account and install Microsoft Authenticator on their mobile phone.

14.3.7 Data Ownership

Windward has to be installed on premises to give users a complete control over their data and the document automation service.

15 Woodpecker

Woodpecker is a document automation tool based on a Microsoft Word Add-In with some cloud functionalities. It is being developed by the US-based startup with the same name which was founded in 2017. The Add-In is run on a local machine but also offers a cloud-based template repository and integration into automated workflows.

15.1 Executive Summary

Woodpecker is a user-friendly Word Add-in tool to help users insert dynamic texts into templates. Except for multiple selectors, all other input types considered in this study are supported by the tool. Woodpecker develops an intuitive user interface for defining logical expressions. The tool can handle compound conditional statements, but customizing the order of precedence among logical expressions is not possible. Besides, basic and advanced mathematical formulas can be defined as a separate field and included in logical expressions.

To generate documents in Woodpecker, first, users create input fields, then insert them into specific position(s) of a Word document. Next, users fill data into the created fields, and finally populate the Word templates with filled inputs. These documents can be exported to PDF using Word.

Woodpecker leverages Microsoft features in version control, sharing documents, managing user groups and defining their read or write access rights. Besides, Two-Factor Authentication is applicable via Microsoft Authenticator Mobile app, providing a more secure option to protect users' templates.

In general, the tool offers a minimalist, aesthetic and intuitive user interface to create, edit and reuse templates. Woodpecker has input validation mechanisms to prevent erroneous data from entering the system. The tool also offers APIs for various document automation features, such as generating documents from external forms or prepare documents on request from external sources and save them on a cloud storage.

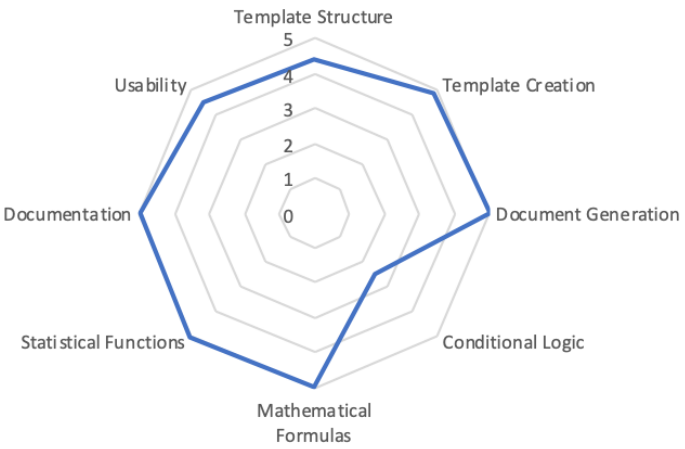

Figure 15.1: Radar diagram for Woodpecker - Core Document Automation Functionality

Integration	✓
Version Control	✓
User Management	
○ User Group Management	✓
○ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
○ Two-Factor Authentication	
◇ OTP via SMS	-
◇ OTP via Email	-
◇ via Mobile App	✓
○ Single Sign-On	✓
Data Ownership	
○ Cloud-based	✓
○ On-premise	-

Table 15.1: Supporting features of Woodpecker

15.2 Evaluation of Core Document Automation Functionality

15.2.1 Template Structure

Woodpecker templates are DOCX files extended with Woodpecker fields. Structuring and formatting of the template is done using Microsoft Word. Fields are defined and managed inside the Add-In and have their value assigned from user inputs. Fields can be used

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	-
Table	✓

Table 15.2: Template structure features of Woodpecker

multiple times inside of a template. The occurrences of a field are displayed inside a document. Field values can be included in the definition of other fields by using a placeholder.

Conditional fields are used to customize the content of a section based on the values of other fields.

Woodpecker templates are flexible and can use most of the required data types, except for multiple selectors.

Score: 4.4/5

15.2.2 Template Creation

A DOCX file can be turned into a template with one click in the Add-In. Woodpecker's machine learning algorithm suggests words and phrases that might be turned into fields. For each suggestion, users can decide whether they want to include or ignore it in the template, or change the field's data type. This feature works best if the underlying text is written in English. Highlighting words or phrases that should be turned into fields or surrounding them with square brackets improves the result significantly.

Besides converting DOCX into templates, Woodpecker can also use existing HotDocs or Contract Express templates. After turning a document into a template the user can add, change and delete field definitions and insert fields into the template.

Field definitions can be reused by copying them to other templates. A string encoding all field definitions can be generated from the Add-In and decoded in the other templates. The copied field definitions are appended at the end of the existing list. There is currently no feature to reuse only a single field definition, text field or structure.

Templates can be uploaded to a cloud-based repository where they can be downloaded, renamed and deleted. They are identified by a unique identifier that Woodpecker attaches to the file. If a new version of a template, even with a different name is uploaded, it will replace the old version in the repository.

While it is intuitive to turn an existing document into a template, Woodpecker lacks version control for templates and allows very limited reusability and central management.

Score: 4.8/5

15.2.3 Document Generation

Template creation and document generation are done in the same user interface. First, users fill in values for all field definitions, then Woodpecker populates the template with the supplied values. Field definitions can be annotated with explanations on their correct usage. The generated document is technically still a Woodpecker template and can be exported by saving it with a new name. Multiple documents from different templates can be generated at the same time using the same user inputs. In this case, the user is prompted to supply additional inputs for field definitions that are not used in the original template.

Generation of a document from a template is done with a minimal number of steps and relies on Microsoft Word for file export. Substituting user inputs with a database to avoid manual user inputs is not possible. Additionally, batch generation from a single template is not possible without using the API.

Score: 5/5

15.2.4 Conditional Logic

Conditional logic is implemented using conditional fields. They can contain any number of conditional expressions which are evaluated from top to bottom. The first conditional expression that evaluates to true decides the value of the field. In case no conditional expression evaluates to true, the field contains a default value or is not displayed at all.

A conditional expression consists of any number of comparisons chained by multiple ”AND” or ”OR” operators. Order-of-precedence cannot be applied to conditional expressions. Operators that can be used in comparisons are ”equals”, ”does not equal”, ”greater than”, ”less than”, ”is empty”, ”is not empty”, ”contains” and ”does not contain”. Numbers and dates may be implicitly converted to strings before being used in comparisons. If data types in a comparison are not compatible, it evaluates to false without displaying an error message.

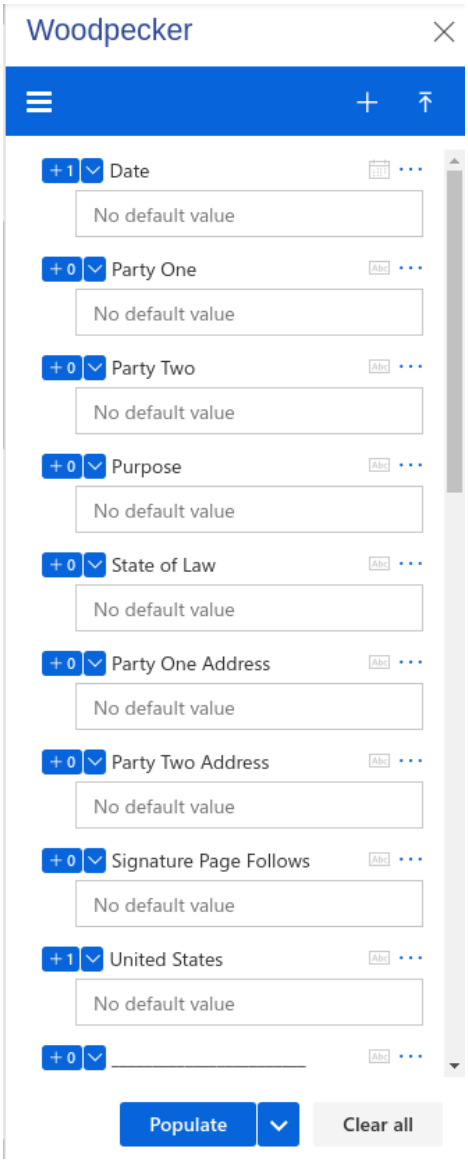

Figure 15.2: Woodpecker - Main View

Woodpecker's conditional logic can handle most common use cases. However, complex conditional structures are hard to validate and are inconvenient to define because users need to declare multiple conditional statements and use several text fields to obtain the correct logical expressions.

Score: 2.5/5

15.2.5 Formulas

Formula fields can be used to dynamically calculate a value using other fields and mathematical operations. This feature uses the operators and functions of Microsoft Excel and extends them with a custom function to manipulate dates. Therefore, formulas in Woodpecker support both basic arithmetic operations and advanced mathematical functions considered in this study. The formula is written in a text field and can include field values by using placeholders. A dropdown menu gives the user an overview of the available fields and functions. If a formula cannot be evaluated correctly, an error message describing the problem instead of a value is displayed. There is no tooltip on the correct usage of functions or the expected inputs and outputs.

Although Woodpecker's formula fields are powerful, users need to search for the correct usage of functions in the documentation of Microsoft Excel.

Basic and Advanced Formulas	:	**5/5**
Statistical Functions	:	**5/5**

15.2.6 Documentation

Woodpecker provides a completely indexed online documentation that is open to the public. All features have articles that describe their usage in detail with explanatory screenshots and short videos. Documentation is only available in English language and is aimed at users without a technical background. The documentation also contains a complete description of the API's endpoints and usages.

Score: 5/5

15.2.7 Usability

The user interface of the Add-In is clean, has low latency and does not feel cluttered. The main view contains an overview of all fields that are frequently used in the template. Fields can be reordered easily and fields that do not require user input can be hidden by toggling the "Simple view" inside the settings. Besides, Woodpecker validates user inputs to prevent erroneous data. Important functions are found where the user expects them and can

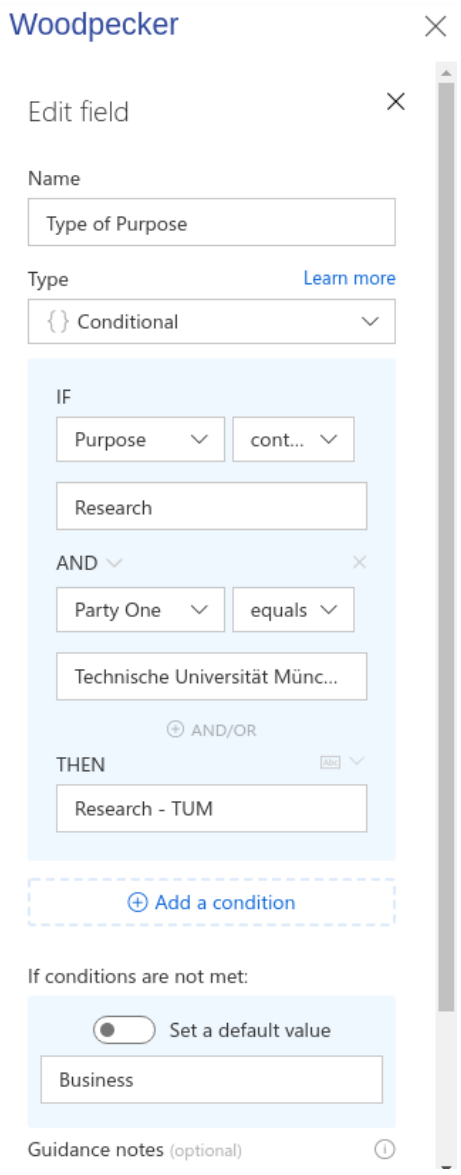

Figure 15.3: Woodpecker - Conditional Logic

be navigated from the central menu that is always available.

Score 4.5/5

15.3 Evaluation of Supporting Features

15.3.1 Integration

Woodpecker provides a RESTful API with three endpoints. They can be used to get all templates in a cloud repository, get a list of all user inputs expected by a template and to generate a document.

Additionally, Woodpecker can be integrated with other web services using Zapier, a web automation platform.

15.3.2 Document Management

Woodpecker does not come with a built-in document management system and has no direct integration with existing document management solutions. However, users can record previous versions of their documents via external cloud-based document management systems.

15.3.3 User Management

Woodpecker does not provide user management capabilities but user accounts can be requested and paid for together. Alternatively, one can leverage Microsoft User Management features with registered Microsoft accounts to share and define access rights to co-authors.

15.3.4 Collaboration

Collaboration in Woodpecker is very limited because each user can only access their own template repository. Users can leverage the sharing templates and more advanced collaboration features provided by Microsoft.

15.3.5 Enterprise Scalability

Scaling with Woodpecker depends on the successful integration with other systems through its API or Zapier, which provide critical functionalities for Woodpecker. If Woodpecker is used as a standalone tool, it cannot address the needs of large organizations. Priority support and user training are provided for enterprise customers.

15.3.6 Advanced Authentication

As a Word plugin, Two-Factor Authentication and Single Sign-On are provided to users via a Microsoft Authenticator Mobile app to access Word documents and use the Woodpecker plugin afterwards.

15.3.7 Data Ownership

If Woodpecker is not integrated with other systems, using the cloud repository is completely optional and all other functionalities can run on a local machine. The repository is isolated from other users even within the same organization, hence sharing templates or documents has to be done outside of Woodpecker. Users can always download, change or delete files from their repository.

16 XpressDox

XpressDox is a document automation tool that runs in the cloud or is installed on premises. It is complemented by a Microsoft Word Add-In to create templates. The tool is developed by the company with the same name since 2008.

16.1 Executive Summary

XpressDox is a tag-based tool for document automation. All of the input types considered in this study, except for multi-selector, are supported by the tool. XpressDox can handle logical expressions at Level 4 complexity (see Section 2.1.4) and several other advanced mathematical formulas.

To generate documents in XpressDox, users insert tags into their templates. Next, when users trigger the document automation process, XpressDox displays a dialog asking users to input values into the tags, and finally generates documents by replacing tags with their corresponding input data.

XpressDox's tag-based mechanism is intuitive for defining dynamic elements only when tags are used as input fields. Applying additional features such as mathematical formulas or logical expressions requires users to search or memorize the corresponding function name or syntax. The tool is able to prevent erroneous data by validating inputs before generating documents.

Collaboration, two-factor authentication, user and access right management features are provided by Microsoft as XpressDox is a Word plugin. The tool offers REST APIs for external systems to use its document automation features.

In general, XpressDox develops a document automation tool by inserting tags into static templates. This mechanism is familiar for advanced users, yet non-technical users can also quickly adapt to the system thanks to the tool's explanatory documentation.

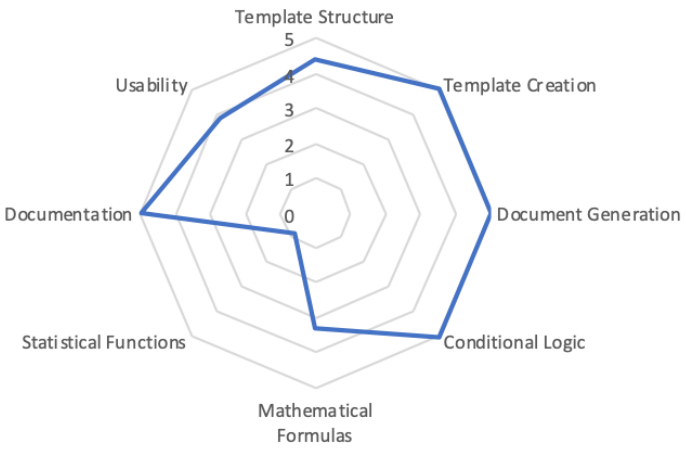

Figure 16.1: Radar diagram for XpressDox - Core Document Automation Functionality

Integration	✓
Version Control	-
User Management	
◦ User Group Management	✓
◦ Access Rights Management	✓
Collaboration	✓
Advanced Authentication	
◦ Two-Factor Authentication	
OTP via SMS	-
OTP via Email	-
via Mobile App	✓
◦ Single Sign-On	✓
Data Ownership	
◦ Cloud-based	✓
◦ On-premise	-

Table 16.1: Supporting features of XpressDox

16.2 Evaluation of Core Document Automation Functionality

16.2.1 Template Structure

Templates in XpressDox are comprised of a DOCX document body annotated with tags that act as functions or input fields. A questionnaire is automatically generated from these annotated tags to collect inputs from users. Table 16.2 illustrates the types of tags can be

Text Label	✓
Text Input	✓
Text Area	✓
Numeric Input	✓
Date (Time) Picker	✓
Single Selector	✓
Multiple Selector	-
Table	✓

Table 16.2: Template structure features of XpressDox

used to capture user inputs.

Tags can also be used to implement conditional logic, calculate values and dynamically change the configuration of the template during the document automation process.

Score: 4.4/5

16.2.2 Template Creation

Templates are created in a Microsoft Word Add-In and can be deployed locally or on an XpressDox server. To create new templates, tags are added to an empty or existing DOCX file at the expected positions in the generated document. Words, phrases or text blocks can be made conditional by placing them between ”If”, ”Else” and ”End” tags (see Section 16.2.4). Templates are managed locally and can include other templates that are locally available by using a tag. The included templates are also generated during the document generation process and inserted into the final document at the position of the tag.

In general, template creation is not complicated but non-technical users need more guidance to familiarize themselves with the declaration of logical expressions, mathematical formulas and other advanced features.

Score: 5/5

16.2.3 Document Generation

The document automation process starts with users answering questionnaires that are automatically generated from tags in a template. Templates can be configured to use a customized questionnaire, generate PDF documents instead of DOCX, and to use external data sources like databases in addition to user inputs. After all questions have been answered, the document is generated and opened in a new window of Microsoft Word where

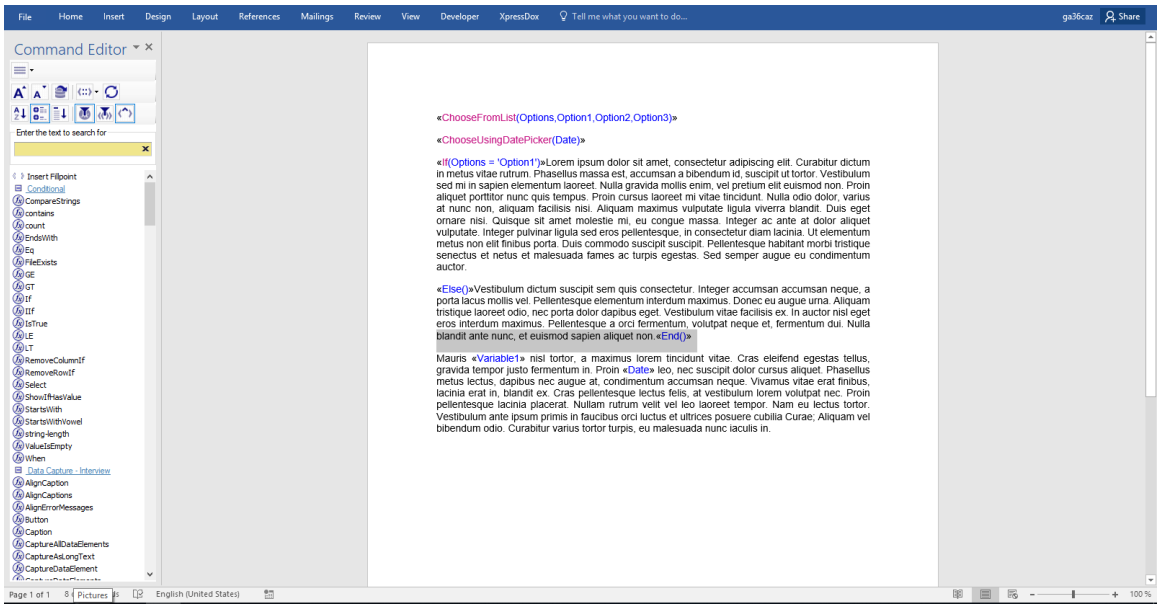

Figure 16.2: XpressDox - Template Structure

it can be edited further and saved.

The document automation process of XpressDox is straight-forward and offers a variety of options for validation. XpressDox supports data sources from MS Excel, MS Access, MS SQL Server, MySQL, and other databases which have an ODBC driver interface.

Score: 5/5

16.2.4 Conditional Logic

Conditional logic can be used inside of a template using ”If”, ”Then” and ”End” tags that contain a conditional expression. A conditional expression consists of comparison(s) between a user input value and another input or static value.

XpressDox supports chained or nested comparisons which are validated before the list of expected user inputs is displayed. In case of a syntax error, the tool displays an error message showing the expression and position in the expression where it occurred. Mathematical formulas can also be added into logical expressions to manipulate input values or provide static mathematical values (see Section 16.2.5).

Support for conditional logic in XpressDox is powerful and easy to validate.

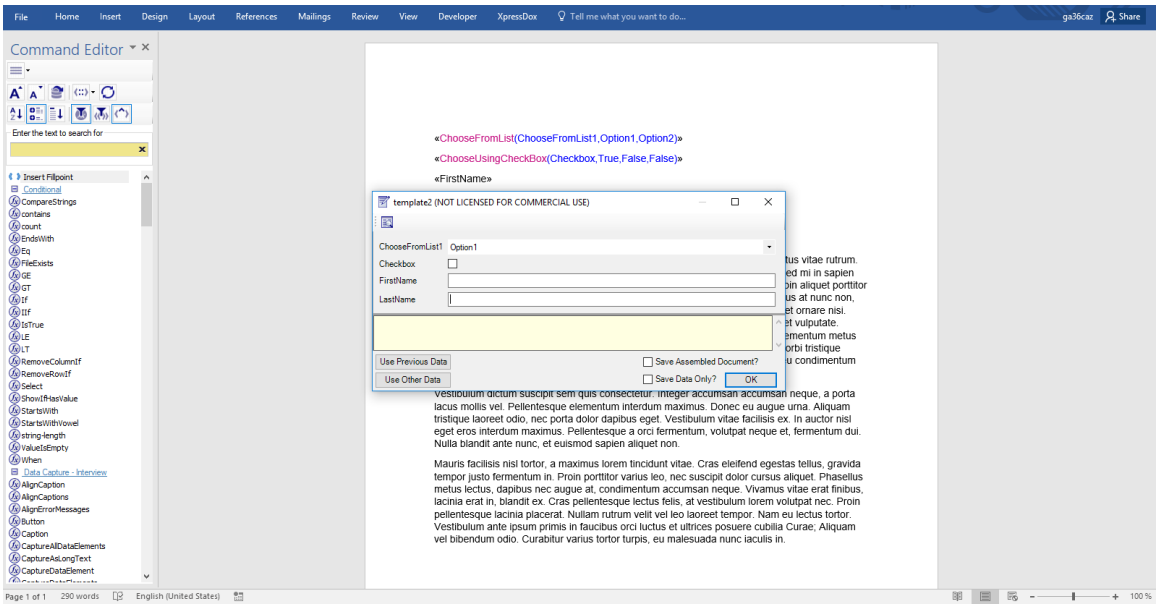

Figure 16.3: XpressDox - Document Generation

Score: 5/5

16.2.5 Formulas

Basic arithmetic calculations can be used inside tags and logical expressions. In case of a syntax error, an error message is displayed, showing the incorrect formula. If an invalid value like a string is used inside a formula, the result of the formula is Not-a-Number. Special tags can be used as formulas to do more complex operations like string manipulation or date functions.

Formulas in XpressDox are flexible and support a wide range of functions, including basic arithmetic operations, exponent, ceiling, and floor.

Basic and Advanced Formulas	:	**3.3/5**
Statistical Functions	:	**0.8/5**

16.2.6 Documentation

The documentation for XpressDox comprises a structured collection of tutorials, articles, and videos explaining features and a detailed list of all available tags. The language is aimed at non-technical users except for the REST API chapters.

Score: 5/5

16.2.7 Usability

The process of modeling templates in XpressDox is not intuitive, especially for non-technical users. In order to use a particular feature, users need to know or remember the names of XpressDox functions so they can search or type them in the template. However, this problem can be solved with the help of the detailed documentation provided in XpressDox.

XpressDox provides meaningful error messages for faulty logical expressions, helping users to locate the source of error. The tool also highlights its functions and macros contained in the template to distinguish them from static and dynamic texts.

Although XpressDox prevents users from certain mistakes, this mechanism does not completely prevent faulty inputs from entering the system. For example, if users accidentally insert alphabet characters into a numeric text field, XpressDox does not show any error message when submitting the form, but records the erroneous data as 0.

Score 3.8/5

16.3 Evaluation of Supporting Features

16.3.1 Integration

The XpressDox server provides REST APIs with endpoints to manage templates and to generate documents from templates. Additionally, XpressDox offers different ways to integrate with the .NET framework.

16.3.2 Document Management

XpressDox does not offer a document management system. However, users can leverage version control features from external cloud-based file management systems.

16.3.3 User Management

XpressDox does not provide user management. Nevertheless, as a Word Add-in, users can share files with others and restrict their read or write access.

16.3.4 Collaboration

Collaboration using XpressDox is limited as there is no feature to share or communicate with other users inside the tool. Users can leverage the sharing templates and more advanced collaboration features provided by Microsoft.

16.3.5 Enterprise Scalability

XpressDox can be easily integrated with other systems or into custom software using the REST API and .NET integration. If the tool is only used as a standalone Add-In the lack of central management and sharing with teammates makes it hard to scale.

16.3.6 Advanced Authentication

As Word Add-in, Two-Factor Authentication and Single Sign-On are possible in XpressDox if users have a Microsoft account and install Microsoft Authenticator on their mobile phone.

16.3.7 Data Ownership

XpressDox can be installed on premises or run in a cloud. All templates and documents are managed manually and stored as DOCX files. Documents can be generated from within the Add-In without uploading any data to a server or the cloud.

17 Conclusion

Thirteen tools were evaluated in this *Document Generation Tool Survey* with thirteen unique approaches for document automation, but a few common themes emerge. There are two major ways that tool vendors have implemented templates. The preference for one type or the other depends on the requirements in the individual case of organizations.

A number of tools use Add-Ins for Microsoft Word to annotate existing DOCX files with custom tags that are used as placeholders or to mark the beginning and end of conditional text blocks. Most users are already familiar with Microsoft Word and large parts of the template creation like formatting are delegated to the underlying editor. At the same this approach places constraints on the design of the user interface and can make it challenging to validate the correct usage of all tags in a template. Other tools are built around a custom editor that has been built from scratch or templates that enforce a high degree of structure. This makes it easier to create templates that adhere to a strict structure but can also limit the number of structures and styles that can be modeled in the template.

Templates are modularized to a varying degree with a higher modularization leading to more reusability between templates. All tools in the survey offer an API to create automated workflows by integrating with other systems that trigger the document generation process. To manually generate documents a questionnaire is usually used to gather user inputs.

An important differentiation of the tools in two categories are the available modes of deployment. While some tools can be run on a local machine or installed on premises, others run in a cloud environment. In the legal domain, a cloud-only approach can be problematic as data related to cases may be protected by company policies or state laws.

Bibliography

[1] L Karl Branting, James C Lester, and Charles B Callaway. Automating judicial document drafting: A discourse-based approach. In *Judicial Applications of Artificial Intelligence*, pages 7–45. Springer, 1998.

[2] Marc Lauritsen and Thomas F Gordon. Toward a general theory of document modeling. In *Proceedings of the 12th International Conference on Artificial Intelligence and Law*, pages 202–211. ACM, 2009.

[3] Marc Lauritsen and Alan Soudakoff. Power tools for document preparation. *See http://www. capstonepractice. com/amlaw6. pdf*, 1998.

[4] Miro Lehtonen, Renaud Petit, Oskari Heinonen, and Greger Lindén. A dynamic user interface for document assembly. In *Proceedings of the 2002 ACM symposium on Document engineering*, pages 134–141. ACM, 2002.

[5] Darryl R Mountain. Disrupting conventional law firm business models using document assembly. *International Journal of Law and Information Technology*, 15(2):170–191, 2006.

[6] Jakob Nielsen. Ten usability heuristics, 2005.

[7] Daniela Tiscornia and Fabrizio Turchi. Formalization of legislative documents based on a functional model. In *International Conference on Artificial Intelligence and Law: Proceedings of the 6 th international conference on Artificial intelligence and law*, volume 30, pages 63–71, 1997.