Test and Evaluation Management Plan

FISMATIC

Version 0.1

May 29, 2020

This document was developed using version 2.1 (released March 2016) of the Test and Evaluation Plan template.  Printed copies of this document may not contain the most recent updates.

Document Revision History

| Version | Publication Date | Revision Description | Section | Author |
| --- | --- | --- | --- | --- |
| 0.1 | 5/29/2020 | Original baseline document | All | Stacie Whitesides |
|  | <Select a date> |  |  |  |
|  | <Select a date> |  |  |  |

Table of Contents

[1.0 System Overview 1](#_Toc41557291)

[2.0 Test Assumptions and Constraints 1](#_Toc41557292)

[2.1 Test Assumptions 1](#_Toc41557293)

[2.2 Test Constraints 1](#_Toc41557294)

[3.0 Test Strategy 1](#_Toc41557295)

[4.0 Test Levels 3](#_Toc41557296)

[4.1 Unit Test Strategy 3](#_Toc41557297)

[4.2 Component Test Strategy 3](#_Toc41557298)

[4.3 Integration Test Strategy 3](#_Toc41557299)

[4.4 System Test Strategy 3](#_Toc41557300)

[4.5 User Acceptance Strategy 3](#_Toc41557301)

[5.0 Security Control Assessment 3](#_Toc41557302)

[6.0 Additional Information 1](#_Toc41557303)

[7.0 Points of Contact 1](#_Toc41557304)

[APPENDIX A Test Methods Guide 2](#_Toc41557305)

[APPENDIX B Test Types Example Guide 3](#_Toc41557306)

[APPENDIX C Acronyms 6](#_Toc41557307)

[APPENDIX D Glossary 7](#_Toc41557308)

[APPENDIX E Referenced Documents 8](#_Toc41557309)

[APPENDIX F Download Requirements and Software Download Instructions 9](#_Toc41557310)

# System Overview

FISMATIC is a proof of concept (PoC) that is currently being developed by the Census Bureau Center for Applied Technology (CAT). This is a machine learning PoC that sets out to assist and streamline the bureau’s Authority to Operate (ATO) with natural language processing. FISMATIC seeks to demonstrate improvements to certifying the security and compliance of bureau information systems by reducing costs, speeding time to completion, and improving consistency. This TEMP outlines the process by which the CAT PoC team will conduct pilot testing.

FISMATIC is a Python program and the interface is a Tkinter GUI. Data for FISMATIC is downloaded from the Risk Management Program System (RMPS), modified, and then loaded into FISMATIC. The tool then organizes the control steps and provides step by step recommendations based on successful ATOs and user input.

Figure 1: FISMATIC System Operation

# Test Assumptions and Constraints

## Test Assumptions

1. Test will be run on a Census Laptop or Desktop because of the sensitivity of the data
2. Secret Agent has been setup and running and will be used to download sensitive ATO data
3. Either Python 3 is installed on the computer, or Anaconda Navigator is installed to allow download and installation of Python 3

## Test Constraints

1. Testing data will be highly sensitive data from previous ATO submissions
2. Tests will consist of a combination of System Testing and User Acceptance Testing
3. Tests will be conducted using tester Census Laptops or Desktops

# Test Strategy

The PoC pilot test model evaluation will enable end users to evaluate real-world scenarios to test the FISMATIC tool before it is deployed. Testing scenarios may include:

* **User Scenarios Testing**—Evaluates the user interface and the quality of the responses presented by the model, use of the output, and provide feedback.
* **Business Process Testing**—Evaluates the system based on real-world business processes and workflows

The PoC team is currently conducting pilot testing across various user groups. The PoC pilot testing is an important activity necessary to verify PoC design assumptions and identify priorities for future capability development. Pilot users have been identified based on knowledge of the ATO process, experience with Office of Information Security (OIS) processes, and National Institute of Standards and Technology (NIST) Special Publication (SP) 800-53.

The focus of the FISMATIC pilot testing are on the following capabilities:

* Development of User Responses
  + Allows the user(s) to edit responses or enter their own response
  + Gives test step definition and status
  + Enables user to save response and continue with next step
* File Upload
  + The user is able to upload CSV file on the front end of FISMATIC

The FISMATIC Test Strategy will include the following elements:

* **Test Workflow** – The PoC pilot test is currently being conducted in an iterative agile iterative. After the PoC team has identified the test users, they are then sent test instructions to begin testing. Pilot testing for the PoC is conducted in two-week increments.

| **Testing Type** | **Scope of Testing** | **Number of Testers** | **Testing Schedule** |
| --- | --- | --- | --- |
| Pilot Testing | Pilot test against the capabilities of the tool | 2-4 | Two-week rotational test |

Table 1: Test Schedule

* **Severity Level** – There will be no severity level for testing the PoC.
* **Test Data** – The data used for the FISMATIC PoC is sensitive ATO data that is downloaded from the Risk Management Program System (RMPS). The download from RMPS is a CSV file that is then cleaned and modified before being loaded into FISMATIC. The tool then organizes the control steps and provides step by step recommendations based on successful ATOs and user input. FISMATIC is a Python program and the interface is a Tkinter GUI. The backend reads the CSV file uploaded by the test users, which is downloaded directly from RMPS for each applicable control step.
* **Test Personnel** – Pilot testing is being conducted by bureau system owners and assessor teams with experience executing successful Authority to Operate (ATOs).
* **Test Environment** – Tests will be conducted on the tester’s Census issued desktop or laptop computer. There will be no special software or hardware needed to support the test.

# Test Levels

The PoC testing being performed will be a combination of systems testing to confirm that the software performs as expected and user testing to both confirm that the system provides value to the user and that the installation instructions provided in Appendix F are sufficient to allow the user to install the software

## Unit Test Strategy

N/A

## Component Test Strategy

N/A

## Integration Test Strategy

N/A

## System Test Strategy

The testers will install the required software on their systems as per the instructions in Appendix F. The testers will then prepare an ATO for their systems by using the system to help them develop text replies for each of the required security controls

## User Acceptance Strategy

The users will access the system for:

* **Usefulness** – does the system assist them in preparing the ATO documentation for their system?
* **Accuracy of the Responses** – are the responses provided by the FISMATIC relevant to the ATO being developed?

# Security Control Assessment

The data used for the FISMATIC PoC is sensitive ATO data that is downloaded from the Risk Management Program System (RMPS). The download from RMPS is a CSV file that is then cleaned and modified before being loaded into FISMATIC. The RMPS data extracts contain highly sensitive data since it depicts how various Census systems implement security controls.

# Additional Information

**Test Case**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **I.D.** | **Test Name** | **Type of Test** | **Instructions or Steps** | **Test Description** | **Expected Result** | **Requirement / Use Case ID** | **Status (P/F)** | **Comments** | **Miscellaneous** |
| T001 | SYS001 | System | 1. Follow steps in Appendix F **Initial Download of FISMATIC**. 2. Download RMPS spreadsheet or create spreadsheet. 3. Follow steps in Appendix F **Preparing your own documentation**. 4. Follow steps in Appendix F **Running FISMATIC the Second Time**. | This test will result in the installation of the FISMATIC software and load the backend with the RMPS ATO historical data | If you run into a “certificate note found” error, make a remedy request for a new Secret Agent certificate. Open Remedy> Network>User Certificates>Request Now click “Add” & Type is “Secret Agent/DBSIGN.” If this doesn’t work, try calling IT.  C:\Users\white614\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\1DBFFEEA.tmp  C:\Users\white614\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\8F9F5F7C.tmp | N/A |  |  |  |
| T002 | UAT01 | User Acceptance | 1. CLICK on each control step inside the family (that is Applicable). The system will show two sample responses you can use and work from, or ignore and write your own response. 2. Hit ‘Submit’ to submit or ‘Skip’ to skip that control. If you want to come back to the set of control families later, you can click ‘RETURN HOME’ and the program will record your progress. Note: If you have already run through the entire control family previously, hitting ‘Submit” will overwrite your previous answer so be careful if running through the same control family twice. 3. Once you have run through all the control families and are done providing the responses you want, click ‘DONE’ to export your spreadsheet and exit the program. Note you MUST hit ‘DONE’ to export the spreadsheet, or your information will be lost. | This test tests the full end-to-end use of the system starting from once it is installed to when the output is complete |  | N/A |  |  |  |

# Points of Contact

| Role | Responsibility | Name | Organization | Phone | Email |
| --- | --- | --- | --- | --- | --- |
| Project Manager & Project Owner | Confirm system developed and available for testers | Robinson Cruz | Center for Applied Technology | 301-763-2053 | Robinson.Cruz@census.gov |

1. Test Methods Guide

|  |  |
| --- | --- |
| Test Methods | Description |
| Static Testing | Involves the examination of the program's code and its associated documentation but does not require the program be executed i.e. code analysis, reviews, walkthroughs, inspections, verification etc. |
| Dynamic Testing | Executing programmed code with a given set of test cases is referred to as dynamic testing i.e. automated test scripts, spreadsheet programs, validation etc. |
| White Box Test | Test on the internal structures or workings of a program, as opposed to the functionality exposed to the end-user. This testing can be applied at the unit, integration and system levels of the software testing process and concerned about the internal perspective of the system and programming skills are used to design test cases. The testers are usually developers working on the project to check the code function and determine the appropriate outputs. |
| Black Box Test | The software is tested as a "black box", examining functionality without any knowledge of internal implementation. The testers are usually not part of the development team and only aware of what the software is supposed to do, not how it does it. |
| Specification Based Test | Aims to test the functionality of software according to the applicable requirements. This level of testing usually requires thorough test cases provided to the tester, who then can simply verify that for a given input, the output value or behavior either the same as the expected value specified in the test case. |
| Visual / GUI Test | Testing by showing someone a problem (or a test failure), rather than just describing it increases clarity and understanding. This type of testing is particularly well suited for environments that deploy agile methods in their development of software, since agile methods require greater communication between testers and developers and collaboration within small teams. |
| Gray Box Test | Test that involves having knowledge of internal data structures and algorithms for purposes of designing tests, while executing those tests at the user or black-box level. The tester is not required to have full access to the software's source code between two modules, where only the interfaces are the only part that system exposed to the test. By knowing the underlying concepts of how the software works, the tester makes better-informed testing choices while testing the software from outside. |

1. Test Types Example Guide

|  |  |
| --- | --- |
| Test Types | Description |
| Load Test | Primarily concerned with testing that the system can continue to operate under a specific load and scale whether that be large quantities of data or a large number of users. |
| Stress Test | A way to test reliability under unexpected or heavy workloads. |
| Stability Test | Specific load or endurance testing that checks to see if the software can continuously function well in or above an acceptable period. |
| Installation Test | Assures that the system installation and that it is working on the hardware. |
| Compatibility Test | Check the [compatibility](about:blank) with other [application software](about:blank), [operating systems](about:blank) or target environments that differ greatly from the original i.e. GUI application intended to be run on the desktop now being required to become a web application on a browser |
| Development Test | Software development process that involves synchronized application of a broad spectrum of defect prevention and detection strategies in order to reduce software development risks, time, and costs. |
| Usability Test | Check if the user interface is easy to use and understand. It is concerned mainly with the use of the application and its usability. |
| Sanity Test | Determines whether it is reasonable to proceed with further testing. |
| Smoke Test | Consists of minimal attempts to operate the software, designed to determine whether there are any basic problems that will prevent it from working at all |
| Regression Test | Focuses on finding defects after a major code change has occurred. It seeks to uncover software regressions, as degraded or lost features, including old bugs that have come back. Regressions occur whenever software functionality that was previously working, correctly, stops working as intended. |
| Recovery Test | Test activity on how well an application is able to recover from crashes, hardware failures and other similar problems. |
| Automated Test | The use of special [software](about:blank) (separate from the software being tested) to control the execution of tests and the comparison of actual outcomes with predicted outcomes. |
| Destructive Test | Attempts to cause the software or a sub-system to fail. It verifies that the software functions properly even when it receives invalid or unexpected inputs, thereby establishing the robustness of input validation and error-management routines. |
| Quality Assurance (Q.A.) | Quality checks that prevents mistakes or defects in the system by avoiding problems when delivering solutions or services to customers. |
| Acceptance Test | Acceptance testing performed by the customers, often in their lab environment on their own hardware also known as user acceptance testing (UAT). |
| User Interface (UI) Test | The process of testing a product’s graphical user interface to ensure it meets its specifications. |
| User Experience (UX) Test | User experience test that involves a person’s behaviors, attitudes, and emotions about using a particular [product](about:blank), [system](about:blank) or service. |
| Security Test | Essential for software that processes confidential data to prevent system intrusion by hackers. |
| Accessibility Test | Compliance test with web accessibility initiative, 508 Test and disabilities act of 1990. |
| Internationalization / Localization Test | General ability of software to be [internationalized or](about:blank) translated into a new language or adapted for a new culture Alpha Test- is simulated or actual operational testing by potential users/customers or an independent test team at the developers’ site. Alpha testing is often employed for off-the-shelf software as a form of internal acceptance testing, before the software goes to beta testing |
| Beta Test | After alpha testing and can be considered a form of external user acceptance testing. Beta version is usually released to a limited audience outside of the programming team known as beta testers. |
| API Test | A type of software testing that involves testing APIs directly and as part of integration testing to determine if they meet expectations for functionality, reliability, performance, and security |
| Concurrent Test | The focus is more on what the performance is like when continuously running with normal input and under normal operation as opposed to stress testing or fuzz testing |
| Conformance Test | Verifies that a product performs according to its specified standards for example compilers are tested to determine whether they meet the recognized standard for that language. |
| Requirement Test | Test against the requirement capabilities of the software |
| Penetration Test | Testing methods which evaluates the security of a computer system or network by simulating an attack from malicious source |
| Platform Test | Techniques that tests an application for portability across different platform i.e. browser compatibility (Internet Explorer, Mozilla Firefox, Safari, etc.) |

1. Acronyms

| Acronym | Definition |
| --- | --- |

|  |  |
| --- | --- |
| AC | Account Management |
| AI | Artiﬁcial Intelligence |
| AU | Auditing Requirements |
| CAT | Center for Applied Technology |
| CIO | Chief Information Officer |
| CM | Configuration Management |
| CSV | Common-separate Value |
| FISMATIC | Federal Information Systems Management Assessment Tool for Intelligent Compliance |
| IA | Identification and Authentication |
| ISSO | Information System Security Officer |
| NIST | National Institute of Standards and Technology |
| OIS | Office of Information Security |
| PIK | Protected Identification Key |
| PoC | Proof of Concept |
| RMPS | Risk Management Program System |
| SC | Plan System and Communications Protection |
| SI | System and Information Integrity |
| SP | Special Publication |
| TEMP | Test and Evaluation Management |

1. Glossary

[Define terms used in the document that may not be familiar to all readers. The document author may add or remove rows in the table as necessary. For assistance with terms, see: ORMPE Project Management Glossary.]

| Term | Definition |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

1. Referenced Documents

[Use this section to provide any references noted within or other useful documents to support this document.]

| Description of Supporting Document or Site | Path or Link |
| --- | --- |
|  |  |
|  |  |
|  |  |

1. Download Requirements and Software Download Instructions

**Download Requirements:**

1. A Census-Issued computer (Desktop or Laptop). The data is sensitive, so it must stay on Census workstations.
2. Download and Set up Secret Agent. This is required to receive the download file for FISMATIC. The zip file will contain sensitive ATO data, thus must be transferred using Census approved file transfers system. For instructions, please follow this directions, here: [https://collab.ecm.census.gov/div/ltso/intranet/Documents/SecretAgentVersion6.2.4UserGuide.pdf](about:blank)

**Note**: If you run into a “certificate note found” error, make a remedy request for a new Secret Agent certificate. Open Remedy> Network>User Certificates>Request Now click “Add” & Type is “Secret Agent/DBSIGN”, if this doesn’t work try calling IT.

1. Anaconda Navigator- This is the Census approved download to get Python 3 installed on your machine. You can download this through the Census mirror here: Please Note: If you have installed Anaconda before, do not click this link unless Anaconda has been fully uninstalled from your Census machine.

**For a windows PC**: http://mirror1.csvd.census.gov/python/repo.continuum.io/archive/Anaconda3-2019.10-Windows-x86\_64.exe

**Initial Download of FISMATIC:**

1. Once all three requirements have been met, email Stacie Whitesides (stacie.whitesides@census.gov) and Sarah Crumling (sarah.g.crumling@census.gov) to let us know and we will email an encrypted zip file (to be unencrypted with Secret Agent) followed by another encrypted file contacting the key.key file.
2. You will receive two emails from Sarah or Stacie with the encrypted files. Follow the directions for Secret Agent linked above (in Step 2) for how to decrypt the files. Make sure to download the files directly onto your Census machine (either into Desktop or Documents folder), NOT a shared drive. The dataset FISMATIC uses is currently sensitive data. Remember where you save the ‘key.key’ file.
3. Once downloaded and decrypted, unzip the FISMATIC file (it should prompt you to do this after download, or if you double-click the zip file)
4. Once unzipped, locate the “run\_FISMATIC” file and double click it.
5. Your command prompt should pop up, and will prompt you to download spaCy (a python package). If it prompts you to continue (y/n) type “y” and hit enter.
6. It may take several minutes while the download completes- this should only happen on the very first time you open the file. After the download, wait a few minutes for the FISMATIC file to run.
7. A window should pop up, prompting you to upload a key. Click ‘Upload Key’ to select the ‘key.key’ file that you decrypted with Secret Agent. Uploading the correct key will have the program continue.
8. After the key is verified, a window should pop up with the Navigation Page- if you don’t see it, check your menu bar at the bottom of your screen for an application with a blank page icon, sometimes it doesn’t pop up to the front on the first time.
9. Run through a control family- adding sample responses, typing your own and testing the ‘SKIP’ and ‘SAVE’ buttons to make sure it runs through without errors. When you are done simply exit out with the ‘X’ in the top right.
10. You’re done! If you’ve reached this point, FISMATIC is successfully installed and you can move on to the next set of instructions! To run FISMATIC again at any point, you just double click “run\_FISMATIC” and it should open up without issue. Otherwise, if you run into any issues- email to the PoC team at cat@census.govto let us know and we will help get it sorted out.

**Preparing your own documentation:**

FISMATIC runs based off a spreadsheet that you provide. This spreadsheet should be available to download from Risk Management Program System (RMPS), however it’s a fairly simple spreadsheet to create if any issues occur. The required columns are:

1. **“ctrl\_step”** The Control Step: In the Format XX-00.00.00.00 (ex: AC-02.00.01.01)

2. **“procedure\_txt**”: The definition of the control step

3. **“status\_type”**: Notes if the control is ‘Component Applicable’, ‘Not Applicable’, ‘Tailored’ or ‘Hybrid’

Column titles should make these exactly; however, extra columns can be added and will not be impacted. Note: Even when downloaded from RMPS, the column headers of these 3 columns will need to be changed and saved as a CSV file)

These column headers and entry formats are very specific, as the code reads them in a particular way to display the information. See the ‘Sample\_ATO.csv’ file for an example of how the file should look. Again, any additional columns in your excel files is fine- the program will leave these alone, and simply add a new column containing your responses and save as a separate file. Make sure it is saved as a .CSV file.

Preparing documentation is a tedious and time-consuming step that will not have to happen in the future. Unfortunately, key features need evaluation before the program can be further fine-tuned. If this part is causing issues, please do not hesitate to reach out to and Sarah Crumling (sarah.g.crumling@census.gov) for assistance in formatting your files and pull any needed information. We appreciate the time and effort you’ve taken to help us improve our product design and features.

**Running FISMATIC the Second Time:**

Now that you have done your initial run to download the programs/ensure the download went smoothly and prepared your file correctly you are ready to run FISMATIC on your own.

1. Double-Click “run\_FISMATIC”

2. Upload your key.key file

3. Once you see the navigation page, click ‘Upload File’ and select your own CSV file (formatted as described above). It may take a minute or two for the program to read in your file and update.

4. The Navigation Page should show a button for all Control Family’s represented in your file- Select which one you would like to start with and click ‘Start’

5. The program will run through each control step inside the family (that is Applicable), showing two sample responses you can use, and work from or ignore and write your own response.

6. Once you are satisfied with your response hit ‘Submit’. If you do not want to provide a response to an answer, simply click ‘Skip’. If you want to come back to the set of control families later, you can click ‘RETURN HOME’ and the program will record your progress. Note: If you have already run through the entire control family previously, hitting ‘Submit” will overwrite your previous answer, be careful if running through the same control family twice.

7. Once you have run through all the control families and are done providing the responses you want, click ‘DONE’ to export your spreadsheet and exit the program. Note you MUST hit ‘DONE’ to export the spreadsheet, or your information will be lost.

**A Few Notes:**

Your finished file will export as “FISMATIC\_MM\_DD\_YYYY\_HH\_MM.csv” (aka ‘FISMATIC\_DATE\_TIME.csv’) once you hit ‘DONE’. It will save to the same location as the rest of the FISMATIC files.

Anytime you run into an issue, or find something not straightforward, take note! These are types of things we are looking to improve! While we have a list of some glaring issues (like having to format your excel documents by hand), you have a different perspective and might find issues we haven’t thought about. Also, any additions you think would improve the program.

Feel free to reach out to the PoC team at cat@census.gov at any point with questions or concerns. We are available to help you at any point in your testing.