Rolling Out Blaise 5 – An Interesting Journey

R. Suresh, Keith Bajura, Matt Boyce, Emily Caron, Lilia Filippenko, Preethi Jayaram, Patty LeBaron, Joe Nofziger, Jean Robinson, Gil Rodriguez, Vorapranee Wickelgren - RTI International

Disclaimer: This paper is simply to report on our findings from the evaluation of Blaise 5. It should not be construed as an endorsement as RTI International does not endorse any software or product.

1. Introduction

RTI uses Statistics Netherlands’ Blaise software (Version 4.8) for several critical survey projects. Given that Blaise 5 is a significant redesign, offering new capabilities for web and mobile, we did a very careful evaluation of its features and adapted our case management systems for CAPI and mobile to work with Blaise 5. This presentation will describe our journey to incorporate Blaise 5 into our survey software toolkit and our experience in using the newer features offered by Blaise 5.

RTI started its evaluation almost two years ago. In the first year, we focused on the instrument proper – ensuring the features we have used in Blaise 4 are still available and working as expected in Blaise 5, and that these features worked correctly in windows, web, mobile-web and mobile-app modes. In the second year, we switched our focus to adapting our case management systems to work with Blaise 5. Somewhere in between we used Blaise 5 web version in production for a project as well.

2. Key Features and Requirements

As with any evaluation of CAI software, we decided to ensure that Blaise 5 had all of the features we needed for a questionnaire and met RTI’s other key requirements.

2.1 Questionnaire Features

We tested the software on a Windows desktop, a web-browser on a desktop and a small device, and on the Android and iOS apps. The results were very promising. The summary results are shown in Figure 2.1. Note that these tests were done using version 5.2.5 and it is possible some issues might have been corrected in later versions.

2.1.1 Single and Multiple responses question

Single and multiple responses questions with radio buttons displayed as expected.

2.1.2 Question with an Image

Images were sized correctly and displayed nicely in all devices. We did not test videos but suspect that they will look fine as well.

2.1.3 Grid Question

Grid questions are always difficult to display on small screens, but Blaise 5 does a decent job with it. On handheld web browsers, in portrait mode it requires us to scroll to the right; on the app versions it truncates/wraps the header so that the grid is fully visible. Given the screen limitations, the questionnaire designer should be creative with the header text so that the resultant display looks better.

2.1.4 Slider Question
Sliders are tricky to display on small devices as well, but Blaise 5 does a good job for the most part. Of the three variations of slider questions, the only issues were that the headers were truncated, or it did not display the number selected but the visual cue was there as part of the slider.

2.1.5 Question with Lengthy Response Options
Lengthy response options wrapped nicely in all devices.

Figure 2.1: Feature evaluations summary. X indicates there were some issues with the display.

<table>
<thead>
<tr>
<th></th>
<th>Radio</th>
<th>Image</th>
<th>Grid</th>
<th>Slider</th>
<th>Lengthy R.O.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Desktop Browser</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Mobile Browser</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Apps Portrait</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>Apps Landscape</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

2.2 Compatible with Blaise 4 source code
Given that we have been using Blaise for a couple of decades, we have developed quite a few modules that are re-used over and over again. We did not want to reprogram everything from scratch. In addition, we need to be able to transition long running projects from Blaise 4 to Blaise 5 without too much effort. Blaise provides a tool to convert Blaise 4 code to Blaise 5 and it really works very well.

2.3 Logic engine needs to be as good or better
Blaise’s logic engine is probably one of the best, if not the best, of all CAI software that’s available on the market. We did not want to give up on any of the features of the logic engine. The logic engine is still pretty much the same as Blaise 4 with one subtle exception. Blaise 4 would allow us to enter a numeric answer that is out of range but Blaise 5 automatically truncates it without prompting the user; this could be problematic.

2.4 Program once, use in all modes/devices
This is the ultimate goal of any good CAI software. Even though we program it only once, it needs to have hooks in it so that response options can be customized based on mode. For example, don’t know/refused options show up in CATI/CAPI mode but not in self-interview modes. In Blaise 5, response options, question text, layout and logic can be customized by mode.

2.4.1 Good user experience on all devices
We have no control over the devices that the respondents use to complete a self-interview over the web. We needed the software to support pretty much any device out there, large and small. Through the use of CSS files, Blaise provides the mechanism for customized user experience for all devices.

2.4.2 Easy deployment
Even if we have the best software in the world, if it took too much effort to deploy it, that sort of defeats the purpose of a CAI software. The initial server configuration takes a bit of time, but once we master its
features, actual deployment is relatively easy for web and mobile apps. It might be useful to make this even more user friendly in the future.

2.5 Cloud
Our clients are starting to require us to deploy the software in the cloud - specifically GovCloud. The software should be deployable in GovCloud. *RTI was able to deploy Blaise 5 in GovCloud with minimal effort.*

2.6 Feature Support
Given that no one can predict all of the needs of the future, we need Blaise to provide hooks or support the use of special features that are critical. For example:

2.6.1 Mechanism for calling custom code & external applications (like alien router/proc)
We need to be able to call external programs for a variety of reasons. It might be that the clients require us to call a software they provide for a specific section, for collecting an online consent form, or for an appointment scheduling program. Regardless, there are situations where we need to call external programs and the CAI software has to support it. *Blaise provides a mechanism for calling external programs. We might need additional features in this regard.*

2.6.2 ACASI and CARI (audio self-interview and recorded interviews)
Twenty years ago, ACASI and CARI were a novelty. Now, it is pretty much a standard feature in good CAI software. The challenge is that we need these features on the apps for small devices, as well. *These features are built into Blaise 5; of course, some of them are yet to be released.*

2.6.3 GPS and other sensors
GPS coordinates have become an invaluable tool for authenticating interviews. In addition, other sensors and gadgets are used to collect bio-specimens, etc. The software should allow us to connect to these external devices and gather the results in a reliable fashion. *We haven’t tested this yet, but the hope is that the feature to call external programs will allow this as well.*

2.7 Audit Trail
RTI uses the audit trail files for salvaging data in case of database corruption, authentication of interviews, and on occasion, for timing of sections. Blaise 5 changed the way audit trails are created and the format is different, as well. *RTI needs to revise its processes and systems to adapt to the new methods and formats for audit trail data.*

2.8 Integrate with RTI’s case management systems
Last but not least, Blaise must work with RTI’s case management systems (CMS) for CATI, CAPI, and handhelds. Even if Blaise develops a case management system in the future, it probably will not include the organization specific features that are critical; hence, it is important that the software can work seamlessly with our case management systems. *As described later in this paper, we have been successful in integrating Blaise 5 with our laptop and handheld CMS systems.*

3. Look and Feel
One of the major changes in Blaise 5 is that the Mode Library has been replaced with the Resource Database. Given that Blaise 5 must accommodate multiple modes and devices, this change is a welcome move. Having said that, given the amount of flexibility offered by the Resource Database to configure and customize the look and feel of the screens in varied environments, the task of setting up the database is daunting and tedious.
RTI initially looked at the Resource Database and worked on creating an RTI template, but soon decided to use the default environment provided by Blaise until the software was truly production ready. Based on the feedback from other organizations, this approach might have saved us time and effort. Even the default configuration provided by Blaise was sufficient for the most part.

3.1 RTI’s default setting
Developers enlisted the help of an in-house User Experience / User Interface expert to review and make suggestions on forming an RTI master page template using one of the provided Blaise 5 templates. The goal was to create a look and feel similar to other RTI data collection systems which had been created based on best practices.

The review resulted in a list of specific items to modify, making the process less overwhelming than coming up with a new template from scratch. The requested modifications were doable, but as many have experienced there is a steep learning curve with the Resource Database and making the various changes took time.

Figure 3.1: RTI’s current master page template.

Examples of some of the Resource Database modifications made:
- Overall page style (color, font tweaks, logo)
- Reformatted and moved Back/Next buttons
- Made updates to table column headers, row selection behavior
- Programmed instrument exit to be done using window ‘x’ instead of special button
- Modified error text size and location
- Modified watermark style and behavior
RTI’s master page template has been tested to work for large and mid-sized screens. A future task is to create a working template for small handheld devices.

3.2 Touch Screen version
We experimented with the touchscreen capabilities of Blaise 5 on an MS Windows laptop. We enlarged the size of radio buttons and checkboxes so that the user can select them easily with their fingertips. We also tried using images for response options and the user simply taps on the image to make selections as shown in Figure 3.2.

A few ergonomic issues were discovered in this process and they are not necessarily Blaise issues. Namely, since it is a touchscreen, the user might think that they can:
- swipe up/down on a screen for scrolling, like one would do on a phone/tablet
- expand or shrink the screen display using their fingers

Furthermore, the virtual keyboard took up more screen space than one would expect and so a physical keyboard might be better.

**Figure 3.2:** “All that apply” question for selecting types of lightbulbs. User can tap on the check boxes or lightbulb images to select the corresponding response.

---

4. Setting Up the Blaise Server Environment

4.1 Network Configuration
Figure 4.1 below shows a general diagram of the network connections required to allow Blaise 5 to host both web surveys and mobile device surveys.
4.2 Server Configuration for Web Surveys
Blaise 5 uses a standard IIS web server to host its web surveys. Aside from installing an SSL certificate, configuration of the website is automatically handled either in the initial installation of the Blaise 5 software or by the application itself. Each web survey gets its own subsite under the default web site as shown in Figure 4.2.
4.3 Server Configuration for Mobile Device Surveys
Blaise 5 uses a single port, 8033 (or any non-standard port), for communication to mobile device apps. This is configured by default as port 8033 using HTTP. For security, this setting should be changed to HTTPS and the appropriate server certificate selected as shown in Figure 4.3 below.

Figure 4.3: Recommended configuration settings for mobile device surveys.
4.4 **Server Firewall Configuration**

If the Windows Firewall is turned on, you will also need to create rules to open ports 443 and 8033 (data). While port 8031 does not need to be opened for internet traffic, it is however used for management between servers in the same “server park” so they can communicate amongst themselves and needs to be opened for this purpose.

4.5 **Lessons Learned**

While the configuration is straightforward, it involved multiple people within our IT department to configure different parts of the network as it involves everything from configuring the server to opening ports to firewall changes. We quickly realized that having access to a computer on the internet which is outside of the organization’s network is crucial for testing connectivity. Also, testing connections over various ports can easily be done using telnet.

5. **Windows version**

5.1 **Adapting to RTI’s Integrated Field Management System**

RTI’s Integrated Field Management System (IFMS) is a standard system to use for CAPI projects on laptops for instruments developed in Blaise, RTI’s Hatteras, and other CAI software. The laptop Case Management System (CMS) is configured to launch the specific CAI software as required by the project. Since Blaise 5 uses a different approach compared to Blaise 4 to install and launch Blaise instruments on laptops, we had to adapt our CMS to support Blaise 5. We tweaked our CMS and upgraded the associated Manipula scripts, and now RTI’s IFMS can support Blaise 4, Blaise 5 and other CAI software as needed. A detailed description of this topic is provided in the paper titled, *Blaise 5 with RTI’s Integrated Field Management System on Field Interviewer Laptops*, in this year’s conference proceedings.

5.2 **Using the Data Entry API for a WPF survey application**

Blaise 5 replaced the alien router concept with an API that allows it to work with external programs and routines. RTI developed a text to speech application that is used for ACASI sections a few years back and we depend on that heavily for some major projects. Blaise provided a sample C# project called SpeechWPF which served as an excellent starting point for us. We were able to modify that project to implement RTI’s text to speech application in the Blaise 5 instrument. This was successfully tested on a prototype project to everyone’s satisfaction. RTI also used the API technique for other functions such as replaying the question audio, playing audio corresponding to question help, and managing the display of onscreen keyboards for particular questions on touchscreen laptops.

6. **Web Version**

6.1 **Web version in the cloud**

RTI conducted four production Blaise 5 web surveys between fall of 2017 and spring of 2018 on behalf of a client. The surveys had a combined list sample of 889 individuals across the United States. The questionnaires were hosted in on two Amazon EC2 servers behind a load balancer and were backed by a SQL Server database.

6.1.1 **Setup and Testing**

In preparation, we initially tested Blaise 5 mock web surveys in an Amazon AWS test environment as well as on local servers. We installed and configured Blaise 5 on the servers. Knowing that our client would be managing the production servers, having our own test EC2 instances gave us the ability to understand the port settings as described in section 4.3, and to provide our client with an informed,
detailed request. In addition, we used these servers to settle on our desired configuration of Blaise services and server parks. Blaise North American Support and Statistics Netherlands were very helpful in responding to our queries during this process.

Installation and configuration of Blaise 5 were done using remote desktop of server instances. In contrast, cases could be loaded into the remote database from an RTI desktop via a VPN connection. The VPN service was provided by our client. Data extraction was done over the same connection.

6.1.2 Administration
Invitations were distributed by email including a link to the survey and unique login credentials. A custom web portal provided login and account management and displayed links to available Blaise surveys per user.

To track the status of unopened/partial/complete for each case, we created indicator variables on different paths in the Blaise questionnaires. A scheduled task regularly exported the variables and updated a control system accordingly.

6.1.3 Outcomes
Each of the four surveys was completed successfully and datasets were delivered. Rarely, a user had a problem with a survey hanging with a page load spinner. Exiting and resuming the questionnaire usually resolved the problem.

These surveys preceded our work on updating instruments in production, described later in this paper. Owing to our emphasis on up-front testing and the short time each of the four surveys was fielded (a few months at most), we avoided such updates.

7. Android App Version

7.1 Adapting to RTI's Mobile Case Management System
Over the years, RTI has developed a powerful set of tools to manage complex surveys that may require tracking cases across multiple waves and varying modes such as mail, in-person, and telephone interviews. Additionally, studies often require that interviewers in the field work offline for extended periods of time which makes using Android tablets running the standalone Blaise app a natural choice. However, in order to make efficient use of this configuration, RTI needed to wrap a comprehensive Case Management System (CMS) around the Blaise Survey engine.

RTI chose an existing CMS created in-house that runs natively on Android devices. This system fully integrates with RTI’s existing backend software and allows for many necessary features such as complex case events and histories, managing related cases such as separate screeners and interviews, locking or deleting cases based on predefined responses or remote input from supervisors or RTI, and the ability for interviewers to record events, such as no response to the doorbell, without needing to enter the survey directly.

Statistics Netherlands provides a straightforward Application Programming Interface (API) that allows for the Blaise data entry app to be invoked from another app. This allowed RTI to quickly and efficiently program our CMS to perform the necessary functions in the Blaise app, such as installing an instrument, downloading cases, starting an interview, and uploading data. Additionally, the Blaise app can return the results of an interview to the calling program, allowing RTI to update the status of a case dynamically based on responses and outcomes of a particular survey. Finally, the Blaise app can be programmed to
return directly to the calling program after every action, meaning interviewers never need to enter the Blaise app manually. This provides a single point of entry and simplifies the interviewer’s workflow.

8. Updating Instruments in Production

As we are all aware, we have to plan for instrument changes after the start of data collection. Updating an instrument in production without impacting the integrity of the instrument is a challenge.

Blaise 4 manipula scripts provided the mechanism for converting the collected data from the old format to the updated format and this was a very powerful, robust and dependable feature. With Blaise 5, there are even more challenges:

- the database can be in SQL databases rather than file-based structures; and this means that one might have to automatically make backups of the existing SQL database tables prior to converting them to the new format
- the instrument might have to be updated while respondents are actively using the web instruments
- offline versions of handheld apps need to be updated, but the data might still come in the old format

RTI has not had enough time to fully evaluate these scenarios, partly because the feature to update instruments in production came later in the year.

9. Features Still Needed and/or to be Tested

As mentioned earlier, we are eagerly awaiting a few more features that are critical to RTI. Most of them are already on the roadmap. In this section, we highlight some of those features.

9.1 CARI
CARI is a key requirement for RTI studies. We have briefly tested the laptop versions, but additional testing is required. We also need this feature on Android apps which is slated to be released in the near future.

9.2 Android Data Entry Client
We need the ability to delete specific cases via the API. This is necessary to remove cases that should no longer be on the device and may contain sensitive information.

We would also like the ability to easily transfer information, such as responses, from a partially completed case to another interviewer. This will allow field staff to stand in for others without the need to start an interview over from the beginning, which can be frustrating for interviewers and respondents alike.

9.3 Instrument Updates
As mentioned earlier, we still need to evaluate this feature in depth.

9.4 Range Checks
As mentioned earlier, Blaise 4 allowed us to enter a numeric answer that is out of range, but Blaise 5 automatically truncates it without prompting the user. So, if a person wanted to enter the value 105 for age, and the age was restricted at 0-99, Blaise 5 would have registered it as 10 without prompting it as an error. This would have huge implications for data correctness.
9.5 Double Key Verification
It seems like a lot of web interviews these days provide the option of hardcopy questionnaires, as well. Therefore, it has become critical to provide a mechanism for double key verification for data entry from a hard copy questionnaire.

9.6 Randomization of Responses and Questions
Randomization of responses is useful so that respondents do not simply select the first few options that are provided. Given that quite a few CAI software provide this feature, we are seeing this requirement come up more frequently lately.

Randomization of questions is also critical for questionnaires that collect attitudes and opinions through a series of questions. If the questions are not randomized, there is a potential that the respondents will not carefully read the questions that are presented later in the series.

9.7 Automated Testing Tool
Testing the instrument has always been a tedious and expensive task. It would be nice if Blaise 5 provided an automated testing tool where the system automatically selects one of the responses for each question and routes through the instrument. The system needs to provide an interface through which the developer can specify values for preloaded variables, restrict the options that can be selected for gate questions, and adjust the probability of selection of some of the options for some of the questions. Such a tool will help reduce the level of effort for testing the instrument and will generate a test database that also would be very useful.

10. Next Steps

The next steps for RTI can be summarized as follows:
- Finalize RTI’s standard version of the Resource Database
- Test ACASI and CARI on all devices
- Test updating instruments after the start of data collection
- Tweak manipula scripts for bringing back audit files from laptops
- Do production testing of Blaise 5 for laptops on a project
- Do production testing of Blaise 5 for Android apps on a project

11. Conclusion

We all have been striving for a CAI software that:
- is feature-rich
- has a powerful logic engine
- provides hooks for external programs
- supports customization for better user experience
- can be programmed once for all modes/devices

Blaise 5 might be the answer we have been looking for. Not all the bells and whistles are in place yet, but there is a solid plan to get them in place in the next few months and that’s comforting. As someone mentioned in one of the conversations, this is just the beginning; there will be requests for more features, more flexibility, and definitely newer modes/devices/technologies/methodologies in the future that have to be accommodated. It is going to be an interesting journey for sure, but we have a great starting point.