Model-Based Testing: Your Key to Better Software
The Testing Dilemma: Speed vs. Quality

There’s no question that when it comes to software development, speed to market is essential. But without quality, your development efforts will be in vain. Your users will soon abandon your application and the poor customer experience will hurt your brand.

Testing is at the heart of both speed and quality. The challenge is to enable rigorous testing that fits into the same sprint, while allowing stakeholders—from testers to business analysts—to stay in alignment and remain flexible. This is a tall order that requires replacing the typically slow, manual and error-prone testing process with a powerful, model-based approach.

A model-based testing approach addresses the key challenges that developers and testers face when they attempt to create better software, faster. These pain points span across the testing lifecycle and include:

- Ambiguous requirements
- Poor test case design and limited coverage
- Waiting for test data
- Unavailability of system components
- No automation

Let’s take a look at each of these problem areas and see how CA Agile Requirements Designer, a model-based testing solution from CA Technologies, can help.


61% of users expect apps to start in 4 seconds or less.**

<table>
<thead>
<tr>
<th>Time (seconds)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3%</td>
</tr>
<tr>
<td>2</td>
<td>12%</td>
</tr>
<tr>
<td>3</td>
<td>24%</td>
</tr>
<tr>
<td>4</td>
<td>22%</td>
</tr>
</tbody>
</table>

And severe issues will drive users away.

<table>
<thead>
<tr>
<th>Action</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Looked for an alternate app</td>
<td>28%</td>
</tr>
<tr>
<td>Stopped using the app</td>
<td>37%</td>
</tr>
<tr>
<td>Uninstalled or removed the app</td>
<td>53%</td>
</tr>
</tbody>
</table>

**61% of users expect apps to start in 4 seconds or less.”

Ambiguous Requirements

More often than not, software challenges are introduced at the very beginning of a project, during the requirements phase. Requirements are often ambiguous and incomplete and usually stored in disparate, static formats. Consequently, test cases are manually derived from incomplete requirements and the problem continues to compound as the development process proceeds. The result is that software often fails to deliver a good customer experience, or even worse, defects are detected later in the development lifecycle where they require far more time and resources to resolve. Because most testing teams operate manually, there is no way to automatically or easily update a test when requirements change. And there is a high probability that your requirements will change.

There is a better way—model requirements as an active flowchart.

Here’s how:

- Build a formal model using CA Agile Requirements Designer that is accessible and understandable to the business analysts that already use VISIO, BPM or other tools.
- Eliminate ambiguity and incompleteness by creating a mathematically precise representation of a system.
- Bring users, business analysts and IT into close alignment with a model that everyone can share, review and use.

Figure 1. A basic flowchart depicting data flowing through a trading system application.
Poor Test Case Design and Limited Coverage

Manual test case design is a time consuming, error-prone process. Poor requirements get translated into poor test cases that lead to design flaws and faulty code. There is no real notion of coverage because testing is usually conducted in an unsystematic and ad hoc fashion, leading to just 10 to 20 percent functional test coverage. Poor test case design also leads to significant over-testing of the same features of the application.

There is an easier way—automatically generate optimal sets of test cases.

Here’s how:

- Generate test cases automatically from the solution’s flowchart when the user stories are created.
- Test up to 100 percent of the specified functionality.
- Link the right data and expected results to the relevant user stories.
- Detect defects earlier and shorten test cycles so that testers can deliver software sooner and with a better customer experience.

Figure 2. An example of generating test cases using CA Agile Requirements Designer.
Waiting for Test Data

The right data is never available when testers need it. Data is not linked to tests and testers have to sift through high-volume, low-variety production data sets, which don’t provide adequate coverage. CA estimates that 20 percent of a software development lifecycle is spent waiting for data. These data constraints force testers to wait for data to become available upstream, which means data is not available in parallel, across teams, projects or releases. Data refreshes can take days or weeks to complete, causing significant delays in testing.

There is another way—test data on demand. Here’s how:

- Automate data mining using CA Agile Requirements Designer to find existing data from multiple back-end systems.
- Use a comprehensive set of combinable data generation functions, system variables and seed lists to create any missing data needed for maximum coverage.
- Include future scenarios, outliers and unexpected results, which are not often found in production data.
- Link data to the test cases that feed directly into the test automation engine and avoid the delays created when the wrong data is delivered.

Figure 3. Unresolved test data linked to stored paths in CA Agile Requirements Designer.
Unavailability of System Components

Environmental and data constraints create further delays as distributed teams sit idle waiting for unavailable system components (such as a production database) or data to become available. There are also delays when testing teams wait for test data to be created and provisioned by a central team, and data is usually not available in parallel. What’s more, many organizations still copy and mask production data to their test environments. These copies are costly to maintain, carry compliance risks and cover only a fraction of the tests that need to be run.

There must be a better way—use self-service data.
Here’s how:

- Use CA Agile Requirements Designer to define within the requirements and test the components that need to be virtualized.
- Use the self-service, on-demand portal to define virtual data needed and receive it automatically.
- Generate realistic virtual data when the optimized test cases are needed.
- Quickly and accurately define request/response pairs directly from a message definition or sampled traffic.

Figure 4. Link virtual end points to test cases in CA Agile Requirements Designer.
No Automation

Managing change is possibly the biggest issue for automation. In an ideal world, you want to automatically generate and maintain optimized, automated tests, that link to the data and the expected results needed to execute them. And when changes are made, you want to avoid going back to existing scripts and editing them.

For most organizations, the challenge is that automated testing frameworks are heavily scripted and the script generation is done manually, as is script maintenance. Some organizations use alternate solutions, such as record/playback or script-less automation frameworks (keywords). However, these approaches still bring you back to manual test case design. Ideally, you want to avoid going back around existing scripts and editing them again, which is simply impossible without an automation generator.

**There is a faster way—use automated testing.**

**Here’s how:**

- Use CA Agile Requirements Designer to automate test case design and test case creation to make exhaustive testing possible.
- Eliminate the time wasted on manual test design and maintenance.
- Keep up with changing user needs easily using an automation generator.
- Maximize the value of existing testing frameworks.
- Build up a library of reusable test components.

---

CA Technologies' testing solutions shift left the testing effort into the design phase, automating otherwise time-consuming test asset creation and maintenance tasks thereafter. CA Agile Requirements Designer is an automated testing and modeling solution, enabling organizations to build and deliver high-quality applications to market faster, at less cost. CA also provides a family of products that help you reach your continuous testing goals.

**CA Testing Solutions**

- **CA Test Data Manager.** Creates all the data needed for testing, matched directly to test cases created in Agile Requirements Designer.

- **CA Application Test.** A collaborative, automated testing solution designed to test and validate modern web and mobile applications, along with the APIs, web services and legacy back-end protocols.

- **CA Service Virtualization.** Creates services needed for testing based on virtual end points modeled in CA Test Case Optimizer.

- **CA Agile Central.** Exports requirements to CA Agile Requirements Designer where they are optimized and converted into the right set of test cases.

---

Figure 5. CA integrated testing process—from requirements to environment and data provisioning, test execution and maintenance.
Together, CA testing solutions enable test teams to work in parallel and achieve rigorous testing within a sprint, even as their requirements are changing. Business analysts can communicate evolving user needs using unambiguous flowcharts, from which they can automatically derive subsequent test assets. This includes optimized test cases, test data, automated tests and virtual data. By eliminating delays created by manual testing and environment constraints and maximizing test coverage, you’ll be able to deliver higher-quality applications on time and within budget.
Customer Success

a.s.r., an insurance and financial services firm was able to map a flowchart model to existing requirements in just four hours.  

The CA Technologies solution will help Rabobank increase the efficiency of their business analysts by 10 percent and testers by more than 30 percent over the next three years.

a.s.r. was able to generate test cases which provided 100 percent coverage derived directly from the model.

A multi national financial service provider reduced test case creation time by 95 percent.

---

3 CA Technologies, “Case Study: Agile Requirements Designer at a.s.r.” 2016
4 Ibid
5 CA Technologies Success Story, “Rabobank improves the customer experience with better apps and faster testing founded on CA Agile Requirements Designer,” 2016
6 Metrics collected from CA Technologies’ implementation experience.
CA Technologies (NASDAQ: CA) creates software that fuels transformation for companies and enables them to seize the opportunities of the application economy. Software is at the heart of every business, in every industry. From planning to development to management and security, CA is working with companies worldwide to change the way we live, transact and communicate—across mobile, private and public cloud, distributed and mainframe environments. Learn more at ca.com.