

How Can I Fit Testing Within a Sprint, Without Compromising Quality?

Generate all the automated tests needed for rigorous testing directly from the requirements and link them to the test data needed to execute them. Provide testers with on-demand, parallel access to production-like systems and avoid the delays created by manual testing and system constraints. When the requirements change, the test assets can be updated automatically, reducing laborious test maintenance and helping test teams keep up with changing user needs.

Executive Summary

Challenge

Traditional testing techniques simply cannot deliver at the speed and quality demanded by continuous delivery. Too slow and manual, these approaches carry over the majority of defects that are introduced by poor quality requirements. What's worse, environmental and data constraints leave testers sitting idle, so that delays mount and an unacceptable number of defects slip through undetected.

Opportunity

If testing is going to keep up with the pace required for continuous delivery, it needs to become more rigorous, more automated and more reactive to change. Adopting a requirements-driven approach can enable testers to generate every test case needed for maximum coverage automatically, while finding or making test data on demand. As test assets are derived directly from the design, they can be updated automatically to reflect changing user needs, delivering quality software on time and within budget.

Benefits

With the right tooling and end-to-end testing process, organizations can drive up quality and efficiency at every stage of the development lifecycle. Testing and business initiatives can be brought into close alignment, detecting defects earlier, where they require far less time and cost to fix. Laborious manual testing efforts can be automated, allowing testers to react to ever-changing user needs and rigorously test software in as little time as a week.¹

Section 1: Testing cannot keep up with Continuous Delivery

Traditional Testing Techniques Can't Keep Up

Traditional testing approaches cannot keep up with the speed of change demanded by continuous delivery application development. Too slow, manual and unsystematic, these approaches allow an unacceptable number of defects to slip through, and organizations face an uphill battle in executing the rigorous testing needed to deliver quality software within a sprint.

Testing: too slow, too many, too unsystematic

Many of the challenges are introduced at the very beginning, during the requirements phase. Requirements typically remain ambiguous and incomplete, and are stored in disparate, static formats. Test cases then have to be manually derived from these incomplete requirements. This is a slow, unsystematic process which usually leads to just 10 to 20 percent functional test coverage. Consequently, defects are detected later in the development lifecycle, where they require far more time and resources to fix, as testing continually rolls over to the next sprint.

Automating test execution alone will not resolve these delays because it usually relies on manual scripting. Additionally, automating execution won't improve test coverage. Test maintenance is another factor that will continue to be a major pain point, as testers must identify the impact of a change across systems and update existing tests by hand. This forces testing to roll over indefinitely to the next sprint because it simply can't keep up with changing business needs.

Data and system constraints create delays

Environmental and data constraints create further delays as distributed teams sit idle waiting for unavailable system components or data to become available. There are also delays when testing teams wait for 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 production data to test environments. These copies are costly to maintain, carry compliance risks and cover only a fraction of the tests that need to be run.

Section 2: A fully integrated, end-to-end testing solution

Rigorously Test Software Within a Sprint

CA Technologies offers a complete suite of testing solutions, which can “shift left” the testing effort into the design phase, automating otherwise time-consuming test asset creation and maintenance tasks thereafter.

Requirements and user stories can be imported and modeled as an active flowchart in CA Agile Requirements Designer. Active flowcharting eliminates ambiguity and reduces defects early, in the design phase, as well as allows organizations to generate the smallest set of automated tests directly from the requirements. When the requirements change, test cases are updated automatically, eliminating the bottlenecks created by manual test maintenance and execution.

CA Test Data Manager generates small, secure test data sets that can be found or created on demand. These data sets are matched directly to the optimized tests as they are created by CA Agile Requirements Designer. In the requirements, testers can further specify the system components that need to be virtualized, and use CA Test Data Manager to generate realistic virtual data to cover every response. This rich data can be injected directly into deployed virtual services in CA Service Virtualization. This technique provides testers with simulated access to the otherwise restricted or unavailable components needed to deliver fully tested software to market earlier and at a lower cost.

The following sections discuss the solution components in CA Agile Requirements Designer, CA Test Data Manager and CA Service Virtualization, which together, can enable the continuous delivery of your software applications.

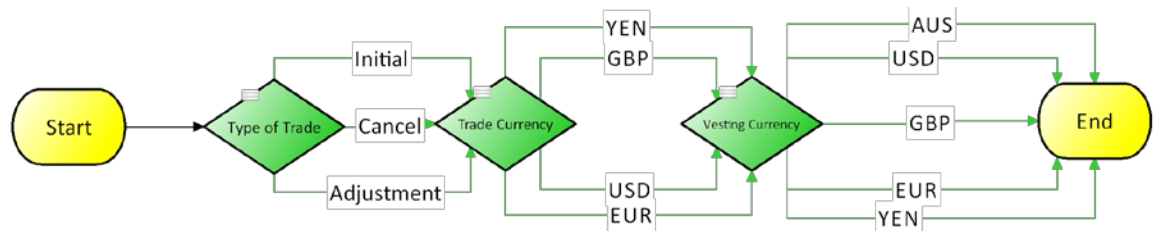
Build quality in from the start

Delivering quality software starts during the requirements phase. Using CA Agile Requirements Designer, existing requirements, user stories and test cases can be imported and converted to an active flowchart model. This brings business and technical teams into close collaboration, providing a single point of reference for testers and business analysts (BAs) alike.

With CA Agile Requirements Designer, user stories can be verified visually with end users. Completeness-checking algorithms can be applied to eliminate the majority of defects that arise from ambiguous requirements and the costly rework they create. CA Agile Requirements Designer will also attach cost and complexity metrics to each requirement and user story, allowing development time and expense to be accurately forecasted before any substantial work begins.

Figure A.

A basic flowchart depicting data flowing through a trading system application.



Using CA Agile Requirements Designer, you can:

- Model requirements as unambiguous, active flowcharts to foster close collaboration between end users, BAs, testers and developers
- Detect requirements defects before any substantial development work begins to avoid the spiraling cost of rework
- Avoid scope creep and budget overrun using accurate cost and complexity metrics to estimate the time and cost of testing and development

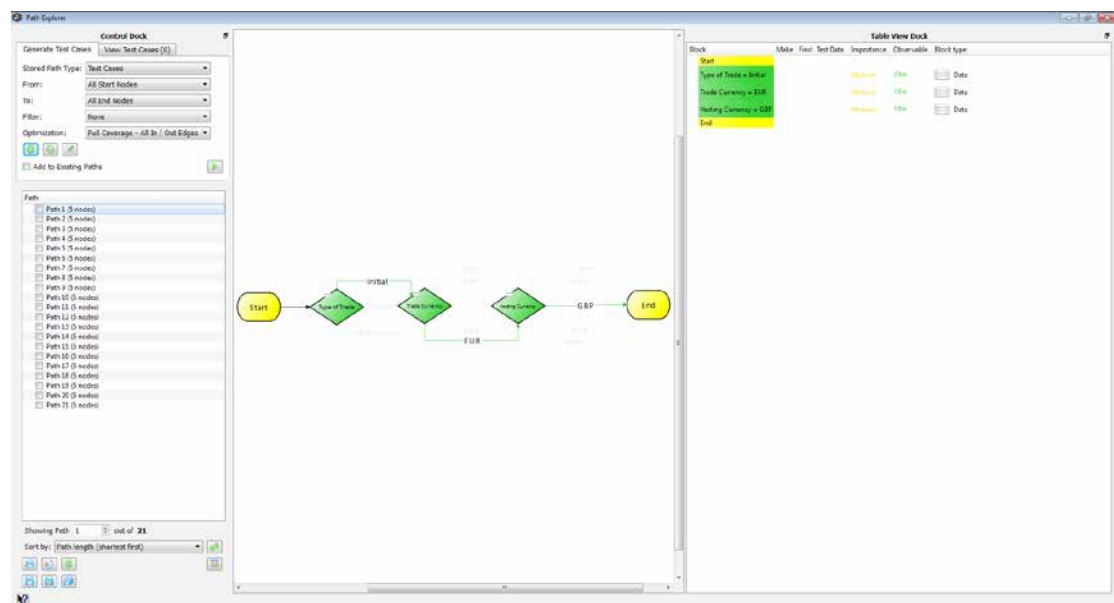
“In CA Agile Requirements Designer, requirements are graphically depicted in a flowchart: a simple, easy way to read and understand a system’s logic, with clear paths through it.”²

Automatically generate the perfect set of test cases

Because the flowchart is a mathematically precise model, you can automatically derive from it the smallest set of test cases needed for maximum coverage. CA Agile Requirements Designer provides multiple optimization algorithms to shorten test cycles, while avoiding the delays created by manual test case design. The optimized test cases can then be pushed out and allocated to available test teams using existing tools such as CA Agile Central or HPE ALM, enabling testers to rigorously test software within the confines of a sprint.

Figure B.

An example of path explorer and test case optimization in CA Agile Requirements Designer.



With automated test case design from CA Agile Requirements Designer, you can:

- “Shift left” the testing effort into the design phase and generate the smallest set of test cases needed for maximum coverage directly from the requirements
- Avoid the delays created by manual test-case design
- Maximize test coverage to shorten test cycles and detect defects earlier, where they require far less time and cost to fix

“This is unique and very welcome functionality and proved invaluable for the project as we only had two weeks to fully test the system.”

– ASR Nederland³

Find and create the right data on demand

Through the two-way integration with CA Test Data Manager and CA Agile Requirements Designer, “test matching” can be used to find or make the data needed to execute the optimized tests as they are created.

Automated data mining will find existing data from multiple back-end systems, while a comprehensive set of combinable data generation functions, system variables and seed lists can be used to create any missing data needed for maximum coverage. This includes the future scenarios, outliers and unexpected results not often found in production data, providing testers with everything they need to rigorously test evolving applications on time and within budget. The data is linked to the exact tests it can run, and can be fed directly into test automation engines, avoiding the delays created when the wrong data is delivered.

Figure C.

Unresolved test data linked to stored paths in CA Agile Requirements Designer.

flow_name	test_name	APPEAL	BALL_HIT	BATTER	BOWLER	NOBALL	PITCHED	PLAYS_SHOT	STRUCKPAD	expected_results
1 LegBeforeWicket	LBW	NO		@randlov(0, @seedlist(Batters)@)@	@randlov(0, @seedlist(Bowlers)@)@					Given Not Out;Not Out
2 LegBeforeWicket	LBW(1)	YES		@randlov(0, @seedlist(Batters)@)@	@randlov(0, @seedlist(Bowlers)@)@	YES				Given Not Out;Out
3 LegBeforeWicket	LBW(2)	YES		@randlov(0, @seedlist(Batters)@)@	@randlov(0, @seedlist(Bowlers)@)@	NO	OUTSIDE LINE			Given Not Out;Out
4 LegBeforeWicket	LBW(3)	YES	YES	@randlov(0, @seedlist(Batters)@)@	@randlov(0, @seedlist(Bowlers)@)@	NO	INLINE			Given Not Out;Out
5 LegBeforeWicket	LBW(4)	YES	NO	@randlov(0, @seedlist(Batters)@)@	@randlov(0, @seedlist(Bowlers)@)@	NO	INLINE		OUTSIDE	Given Out;Not Out
6 LegBeforeWicket	LBW(5)	YES	NO	@randlov(0, @seedlist(Batters)@)@	@randlov(0, @seedlist(Bowlers)@)@	NO	INLINE	NO	INLINE	Given Out;Out
7 LegBeforeWicket	LBW(6)	YES	NO	@randlov(0, @seedlist(Batters)@)@	@randlov(0, @seedlist(Bowlers)@)@	NO	INLINE	YES	INLINE	Given Not Out;Out

Colour Legend

- From Test Data
- From Parameters
- From Decision Tables
- From Subflow Data
- From Made Data
- From Found Data

Buttons: Resolve, Export to CSV, Export to Repository, Remove Selected Paths, Select Matching Paths

Data is stored as reusable assets in a central Test Data Warehouse, where it can be provisioned on demand and in parallel. Dynamic form building means that testers can request data based on exact criteria, receiving it in minutes from a self-service Web portal. As it is provisioned, data is cloned and sophisticated version controls will update data to reflect any changes made to the requirements. Distributed test teams working across versions and releases are provided with the up-to-date data they need, without wasting time waiting for data or finding and building it by hand.

Integrating CA Agile Requirements Designer with CA Test Data Manager allows you to:

- Automatically find or make test data based on the tests which need to be run
- Test every possible outlier, unexpected result and negative scenario to detect defects earlier, where they require far less time and cost to fix
- Provision data on demand and in parallel to avoid the delays created by data constraints
- Significantly reduce the time and resources required to provision data

“With CA Test Data Manager Solution, we reduced our time to get test data from 21 days to two days, making our entire software delivery cycle more efficient.”

— Director of quality assurance and strategy, Financial Services Company⁴

Automated test execution

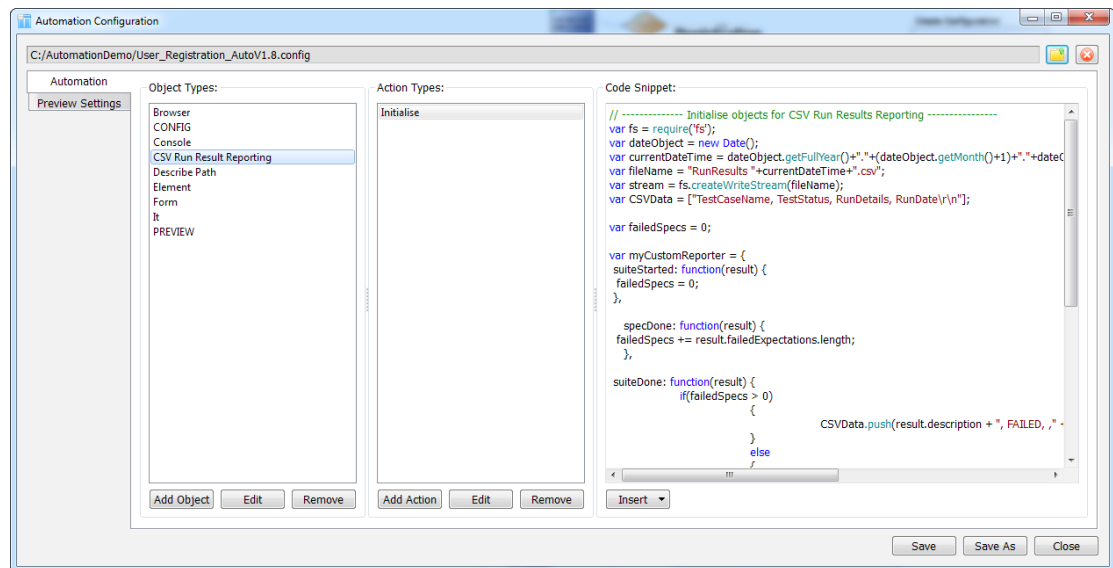
When tests are executed manually, testing can rarely keep up with changing user needs. Likewise, writing each automated test script in a linear manner is highly time consuming. CA Agile Requirements Designer offers two customizable techniques to automatically generate test scripts from the optimized set of test cases created.

The first leverages the Critical Logic TMX framework, which enables users to create a configuration file for potentially any script or keyword-driven automation framework, building up a repository of reusable test objects. Alternatively, code snippets can be overlaid directly into the nodes of the flowchart itself. As the optimized tests are generated, the reusable code snippets specified are then compiled into executable tests. Merged scripts can also be created, to quickly execute each test one after the other.

The flexibility of these approaches means that scripts can be created for practically any existing script-based or keyword-driven framework, without the bottlenecks and poor coverage created by manually writing each script in a linear fashion. The individual test steps can further be shared and re-used from the repository, quickly assembling a new set of optimized, automated tests. Testing becomes both automated and optimized, so that the time spent preparing and maintaining tests does not outweigh the time saved executing them.

Figure D.

A configuration file and compiled script in CA Agile Requirements Designer.



Automating the test automation process allows you to:

- Automatically generate and execute every automated test needed to rigorously test an application, and link it to test data and expected results
- Significantly reduce the time and effort of manual testing and script development with scriptless automation
- Achieve rigorous testing within the confines of a sprint by moving directly from requirements to automated testing

“After two business days, we had executed all 137 test scripts with 100 percent coverage and zero defects found. A result to be proud of.”

– ASR Nederland⁵

On-demand access to production-like systems

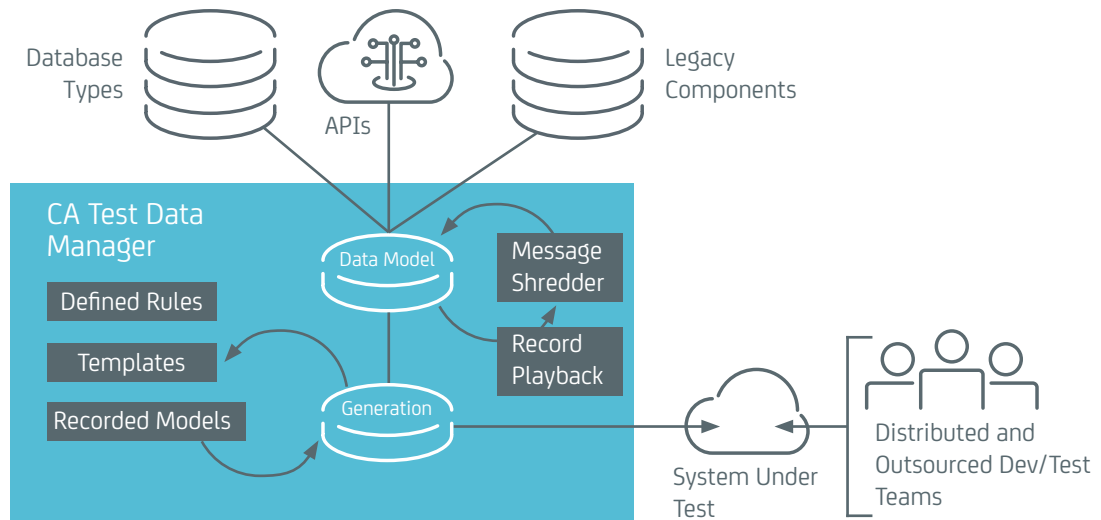
Once testers have the necessary optimized tests and data, they need the environments in which to execute them. Together, CA Agile Requirements Designer, CA Test Data Manager and CA Service Virtualization enable the components of a system needed for complete testing to be created quickly, eliminating the delays associated with system constraints.

Creating realistic virtual services requires rich virtual data, which should be made available, on demand, to avoid testing bottlenecks. Using CA Agile Requirements Designer, testers can overlay virtual endpoints directly onto requirements and test cases, specifying exactly what in a system needs to be virtualized. CA Test Data Manager can then be used to generate all the realistic data needed to execute each optimized test, injecting it into a deployed virtual service in CA Service Virtualization. The automated tests created in

CA Agile Requirements Designer can then be pushed out to CA Application Test, where they can be executed in the deployed service.

Figure E.

Virtual data is generated on the basis of a message definition and injected into a deployed virtual service.



A covered set of virtual data, which includes every possible response, can also be generated using CA Test Data Manager on the basis of a data model created in CA Agile Requirements Designer. This rich test bed of virtual data can be exposed to teams on demand from the web-based Test Data on Demand portal, providing testers with constant access to the request-response pairs they need to execute any possible test. Distributed teams no longer have to wait for unavailable components to become available or be created, helping them cost-effectively deliver fully tested applications to market earlier.

Using the integration between CA Test Data Manager, CA Agile Requirements Designer and CA Service Virtualization, you can:

- Provide testers with on-demand access to environments in which to execute every possible test
- Enable parallel access to interdependent services and components to eliminate the delays created by cross-system dependencies and constraints

“... you have an integrated environment where data is not siloed and therefore closer to both developers and testers.”

– Philip Howard, Bloor Research⁶

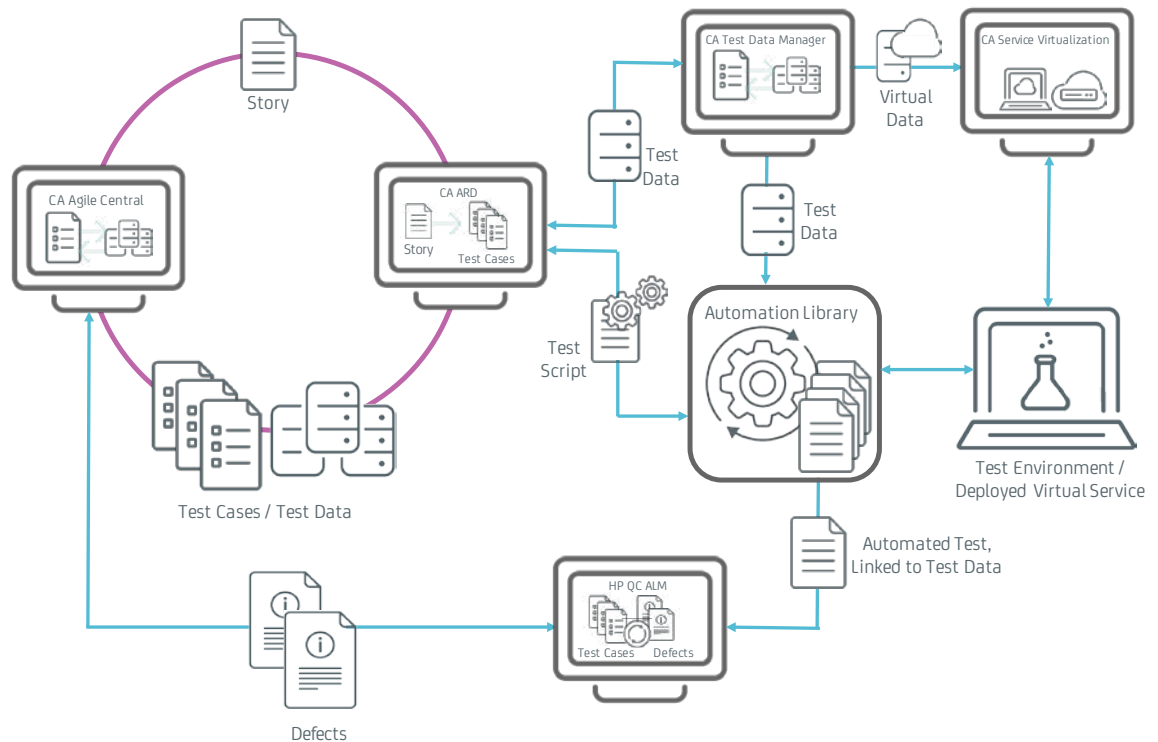
Keep up with constantly changing requirements

With active requirements driving test case design, data creation, test automation and virtualization, all subsequent test assets can be updated automatically when the requirements change.

Using CA Agile Requirements Designer, the potential impact of a change across a system will be identified automatically, with the tests needed to fully test a system that has been updated or created. This eliminates the time wasted on test case and script maintenance, while any broken or redundant tests will be removed or repaired to avoid redundant over-testing. With subflows, changes made to lower-level components will be automatically reflected in master processes, identifying when the integrity of a system might be at risk and validating that a change has successfully rippled up and down a system.

Figure F.

The integrated testing process—from requirements to environment and data provisioning, test execution and maintenance.



Accurate cost and complexity forecasts allow BAs to weigh the impact of a change on testing time and cost. The impact analysis and the built-in entity diagram mapper further allow organizations to visually explore the dependencies between components and test cases in a system. BAs can make informed decisions about which changes are most worthwhile, while test teams can automatically maintain the assets they need to keep up with changing business needs.

By deriving test assets from active requirements, you can:

- Automatically identify the impact of a change system-wide and know exactly which components need to be retested
- Drastically reduce the risk of scope creep and delays by using accurate cost and complexity metrics to forecast the time and cost of change requests
- Automatically update tests when a change is made to requirements and eliminate the bottlenecks created by manual test maintenance

“Future changes to requirements or processes can be easily supported, with an immediate understanding of their impact. This allows testers and the business to stay in control of constantly changing requirements with ease.”

– ASR Nederland⁷

Section 3: Benefits of adopting a requirements-driven, automated approach to testing

Quality Software Delivered Earlier and at Less Cost

Together, these CA solutions enable test teams to work in parallel and achieve rigorous testing within a sprint, even as their requirements constantly change. BAs 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 can deliver higher-quality applications on time and within budget.

Section 4:

Next Steps

To find out more about how the CA Technologies' solutions combine to offer a complete testing suite, please read the Bloor Research White Paper, “[CA Technologies Agile Testing Solutions](#).”



Connect with CA Technologies at ca.com



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.

- 1, 2, 3, 5, 7 *CA Agile Requirements Designer at a.s.r.*, retrieved from <http://www.ca.com/us/collateral/case-studies/ca-agile-requirements-designer-at-asr.html> on the 23rd of February, 2016.
- 4 *The Total Economic Impact of the CA Technologies Test Data Manager Solution*, a commissioned study conducted by Forrester Consulting on behalf of CA Technologies, December 2015. Retrieved from <https://www.ca.com/us/collateral/industry-analyst-report/the-total-economic-impact-of-the-ca-technologies-test-data-manager-solution.register.html>
- 6 *CA Technologies Agile Testing Solutions* (Bloor Research: 2016), retrieved from <https://www.ca.com/us/register/forms/collateral/bloor-research-white-paper-ca-technologies-agile-testing-solutions.aspx> on the 23rd of February, 2016.