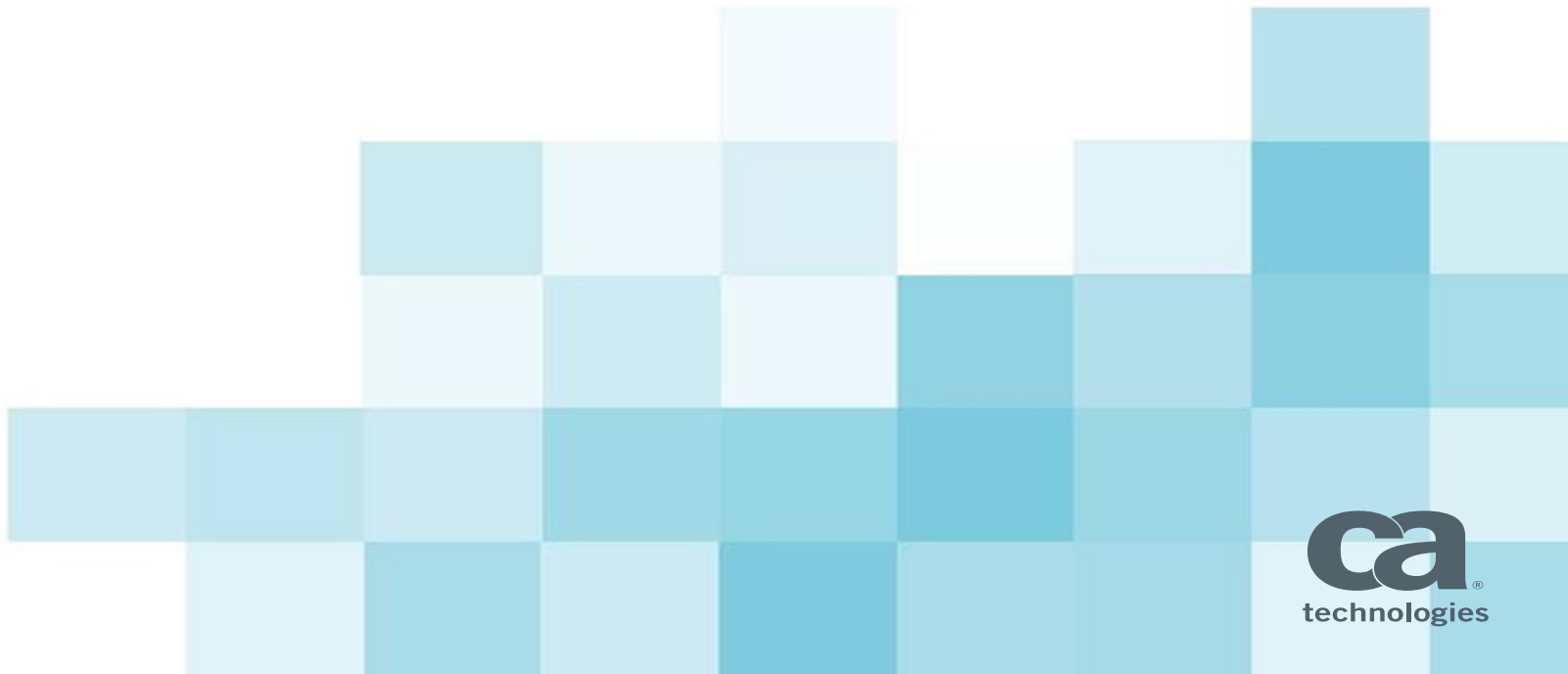


# CA Release Automation Community Webcast Series

Creating Test Cases with CA Agile Requirements  
Designer

Walter Guerrero

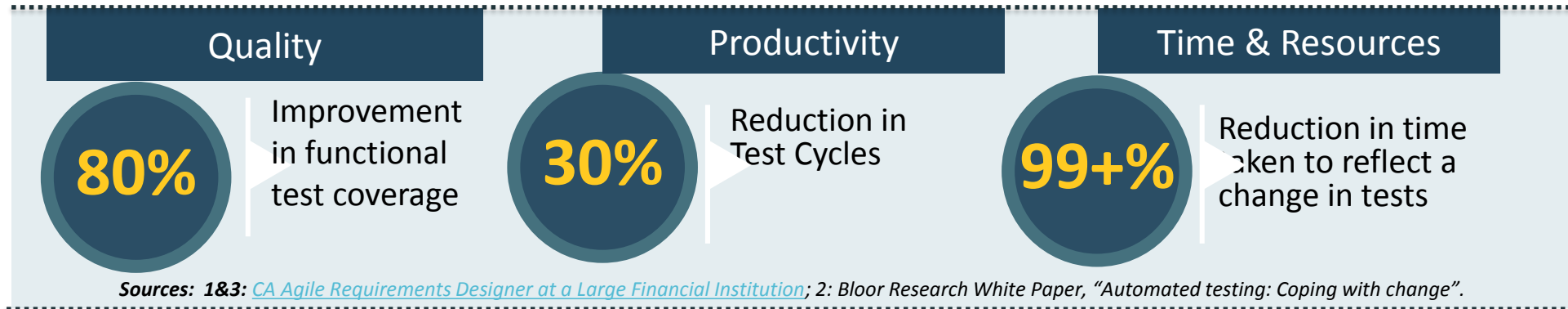
July 26, 2017



# What We'll Cover in Today's Webcast

- What is Agile Requirements Designer
- ARD architecture
- Setup a test case


# Why CA Agile Requirements Designer?



**"In a little over a week, all the processes had been documented, with the accompanying test cases generated, and all with 100 percent test coverage. All test cases had been executed within just three days of further work."**

Test Data Engineer, a.s.r



**30+%** **Expected increased in tester efficiency at Rabobank**

**"CA Agile Requirements Designer met 90 percent of our requirements. At first we didn't believe the solution could really exist!"**

Marcel Mersie, Project & Test Manager, Rabobank

# Why Worry about Test Case Design & Management?

Inaccurate Requirements/Faulty Apps

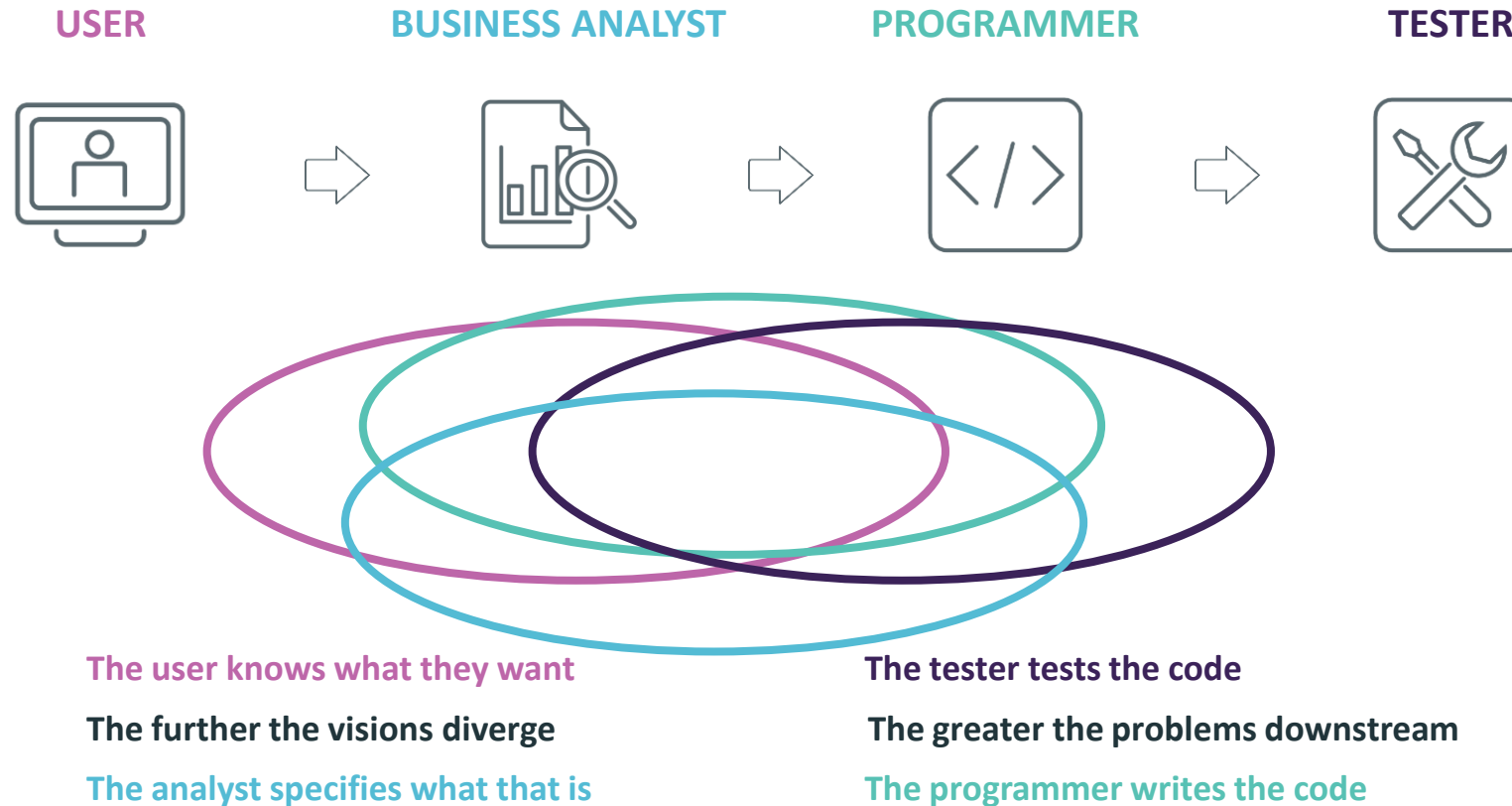
Over testing & Under Test Coverage

Customer satisfaction/quality

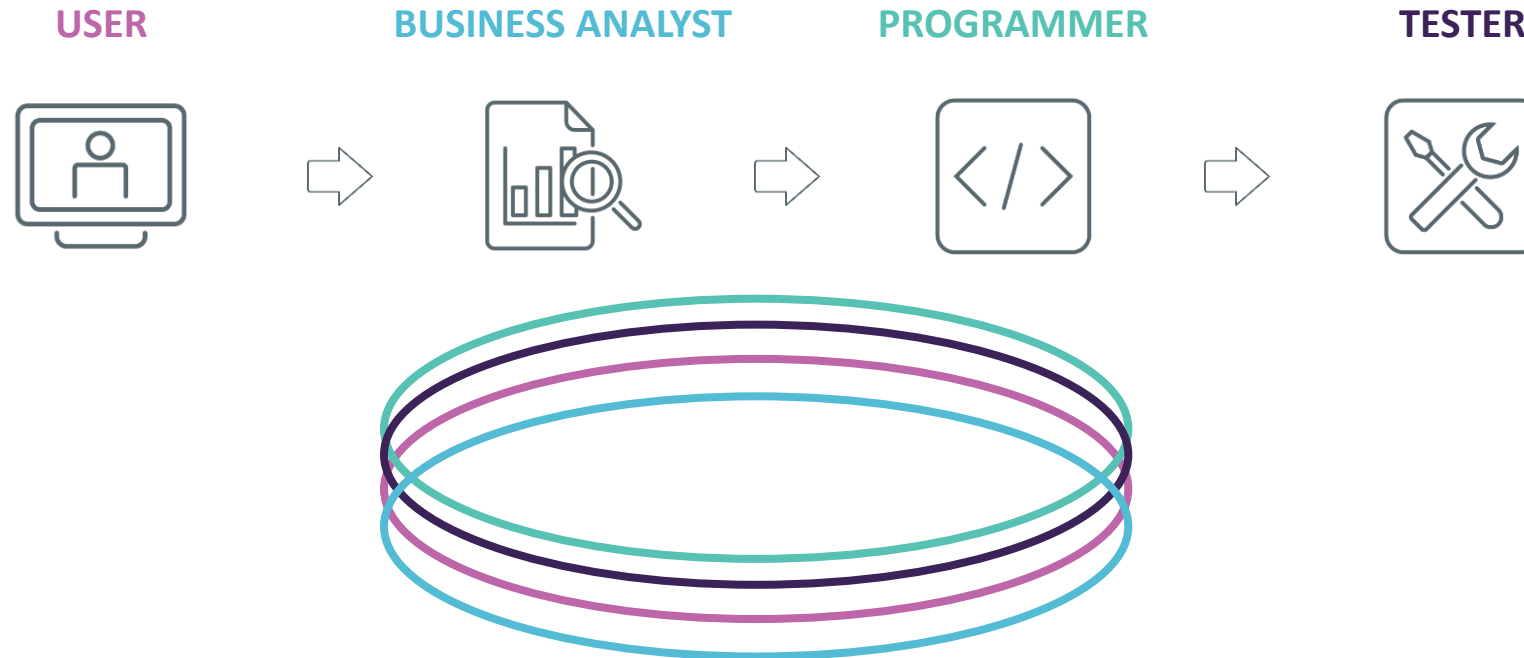
Skyrocketing costs/defects



# Lack of Clarity and Vision During Development



# Clarity and Vision During Development



**The more aligned the vision, the higher the quality app.  
Fewer bugs and faster product delivery.**

# CA Agile Requirement Designer

*Achieve Always On Testing which continuously adapts to changing user requirements*

**All TestDev assets from a single tool**

Create and maintain requirements, user stories, test cases, test scripts and data – all from one tool and flowchart!

**56%**

of software defects can be traced back to the requirements<sup>1</sup>



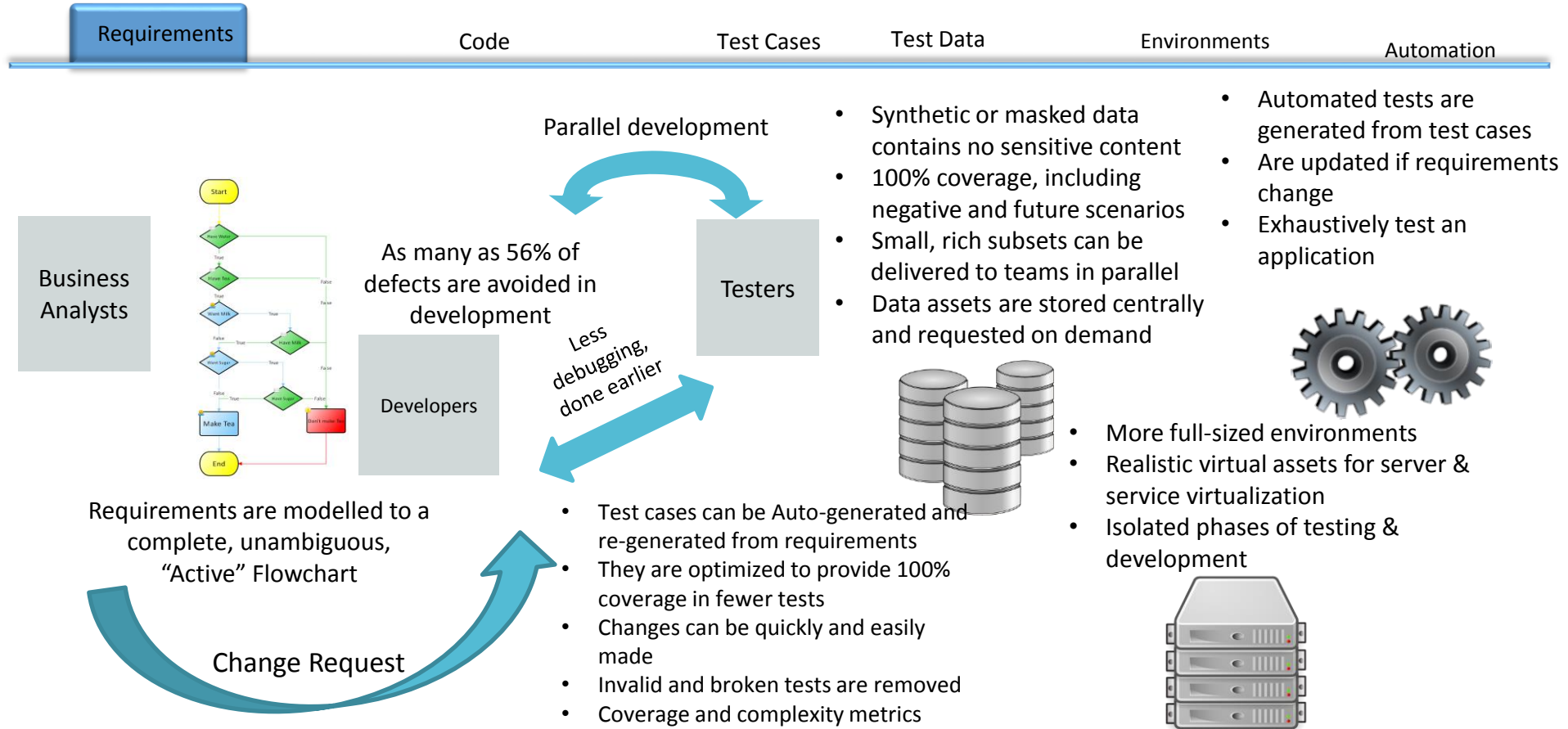
## REAL WORLD RESULTS:

- An **80% average increase** in test coverage
- A **99% reduction** in test maintenance time
- A **95% reduction** in test data provisioning time

## A large European Bank

- Reduced total testing time by **60%**
- One team modelled a system with **28,000 tests** and optimized this down to **39 automated tests**

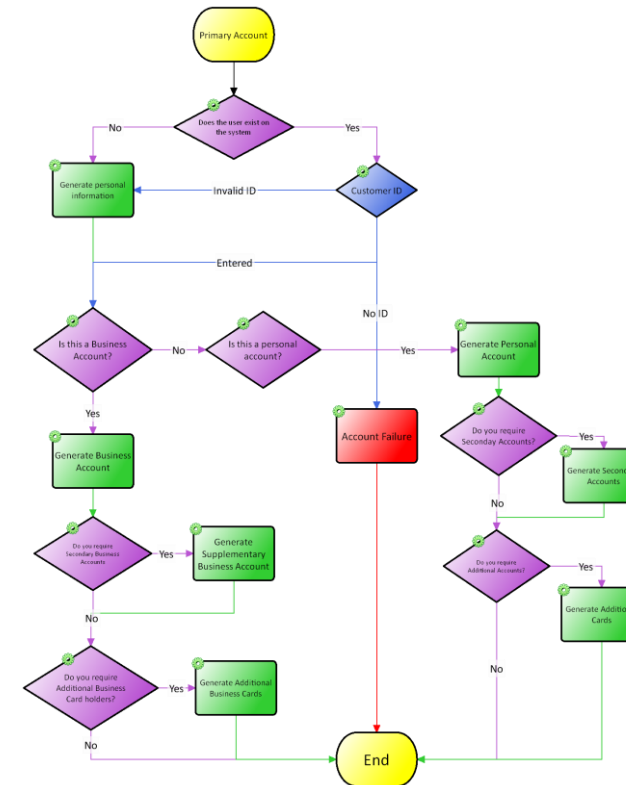
# Introducing a New Approach to End-to-End Testing





# Model Based Testing - Agile Requirements Designer

- Improves communication between IT & Business
- Provides common language and visual flows to validate requirements
- Handles changing requirements to support business
- Automates Creation of the Optimized set of Test Cases
- Documents Test Coverage and Risk
- Identifies impact of changes to existing Test Cases and automates their repair
- Identifies data requirements to support Test Cases for both positive and negative testing
- Generates test data on demand
- Builds and maintains test automation scripts



# Model based testing

Model-based testing lets you define the behavior of a system under test – in other words what is supposed to happen

## Most testing is:

- Random
- Unstructured
- Repetitive
- Not thorough
- Non measurable
- Lagging

## Model based testing is:

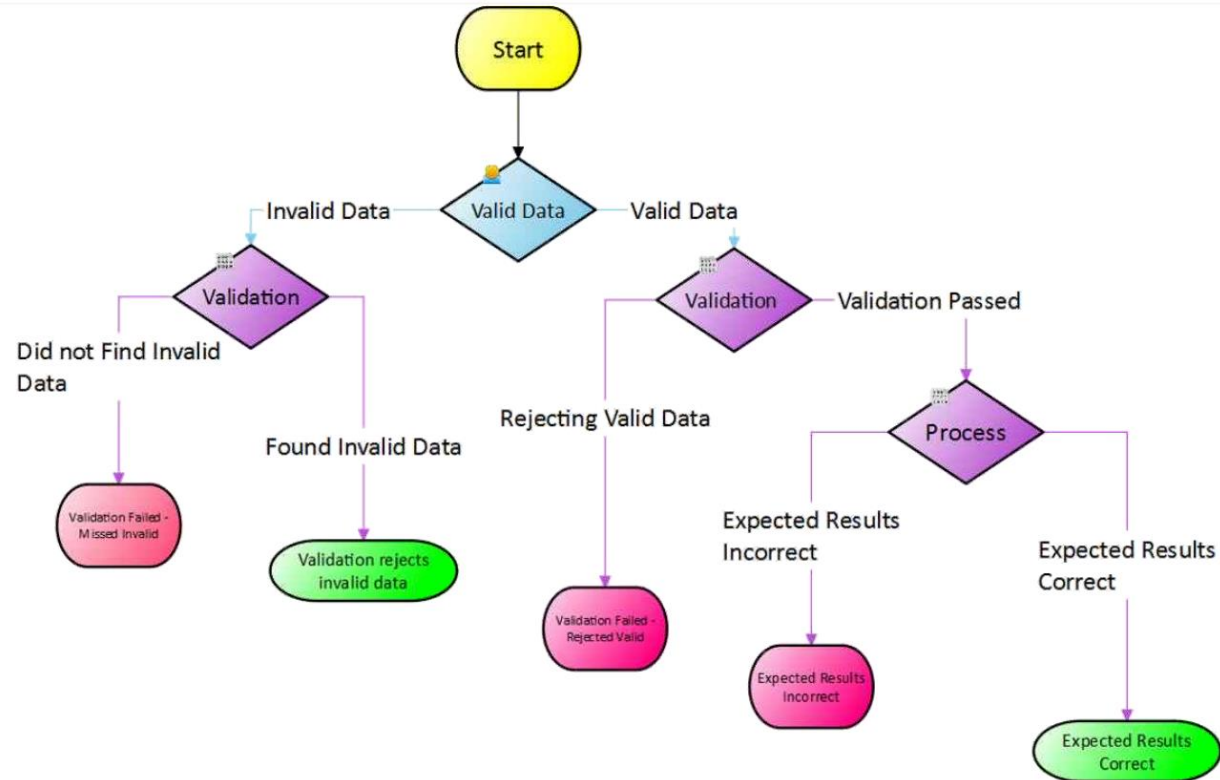
- Accurate
- Structured
- Thorough
- Measureable
- Agile

# Common Challenges in Software Testing

- **56%** of all defects tracked to ambiguities in requirements
- **40x** more expensive to resolve in production
- **10-20%** Test case coverage averages
- **40x** over testing
- **60%** change requests cost time for development & testing

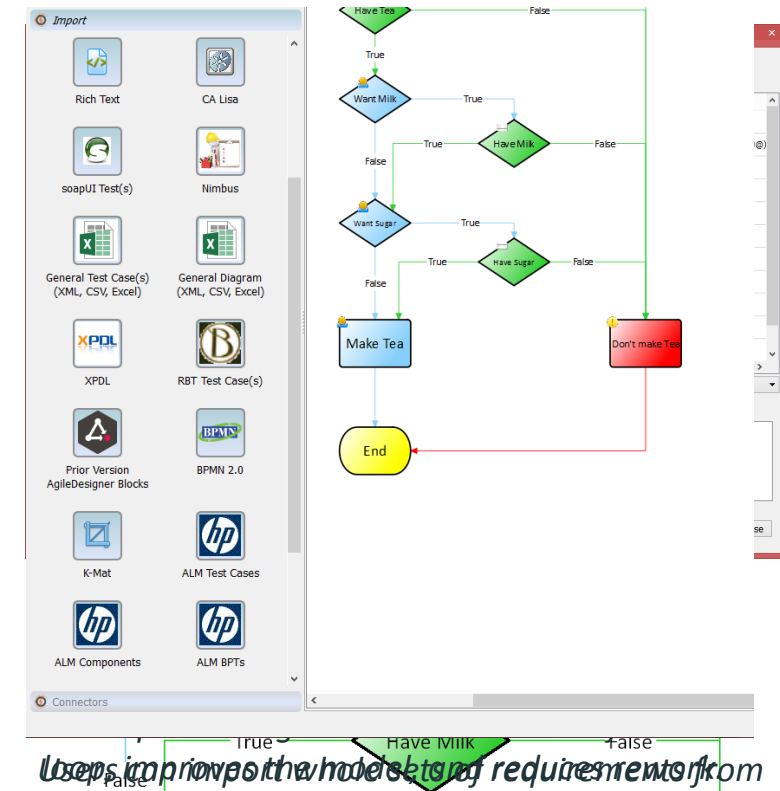


# Minimize Ambiguities with Model -Based Testing



# Agile Requirements Designer – Build Better Requirements

- Map requirements to an unambiguous, active flowchart
- Improve clarity of vision between the business and IT
- Import existing requirements and test cases to an unambiguous flow



Use to improve the whole set of requirements from existing tools, or can drag and drop individual requirements and use cases.

# Create perfect test cases

- Generate the smallest number of test cases with **maximum coverage**
- Test more functionality in **fewer tests**
- **Measure test coverage** and know that every requirement has been tested

Cost and Complexity

Diagram Statistics

Number of Process blocks:	2	Number of Decision blocks:	6
Number of Unconnected blocks:	0	Number of Unconnected Decision blocks:	0
Number of Expected Results:	2	Number of Reached Expected Results:	2

Stored Paths total complexity

Current complexity:	62	Prior Version Complexity:	0
Difference with Prior Version:	62	Percent difference:	---
Current complexity With Virtualization:	44	Current Number of Nodes and Edges:	21
Prior Number of Nodes and Edges:	0	Number of Steps Difference:	0.00%
Current cost:	410	Previous cost:	0
Cost difference:	410	Percent cost difference:	---

Coverage Statistics

Nodes Coverage:	10 / 10 = 100%	More	Edges Coverage:	11 / 15 = 73%	More
In / Out Edge Coverage:	11 / 20 = 55%	More	All Pairs Coverage:	43 / 66 = 65%	More
Overtesting Coverage:	15 / 20 = 75%	More	Export Coverage		
Coverage Explained					

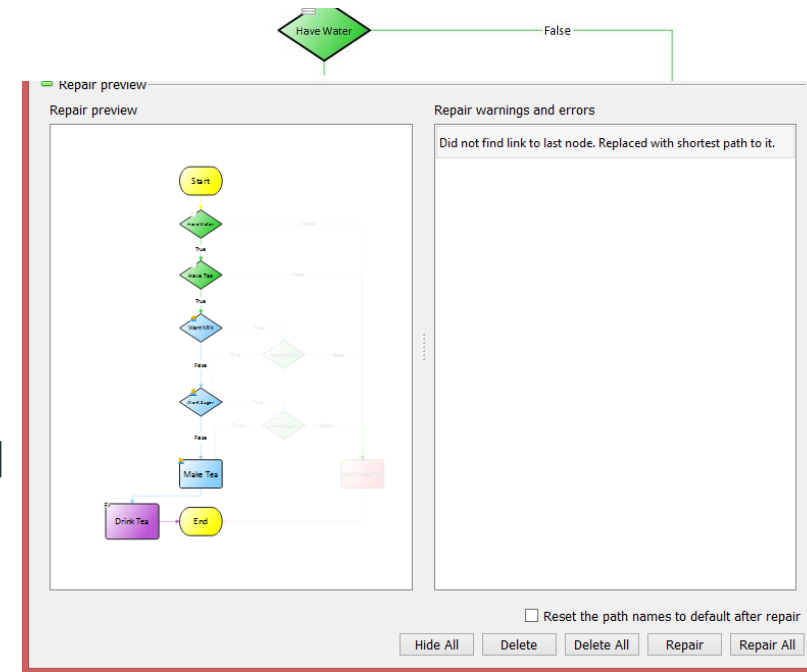
OK

The post-exploration tool in CA Test Case Optimizer

At a financial services, it took 6 hours to create 11 test cases with 16% coverage. It took CA Test Case Optimizer 2 hours to create 17 test cases with 100% coverage.

# Agile Requirements Designer – Auto-update Test Cases & Scripts after Changes

- Simply add a new piece of functional logic to your flowchart
- Identify any broken paths, affected by the change
- Restore 100% functional coverage and produce the new test cases required

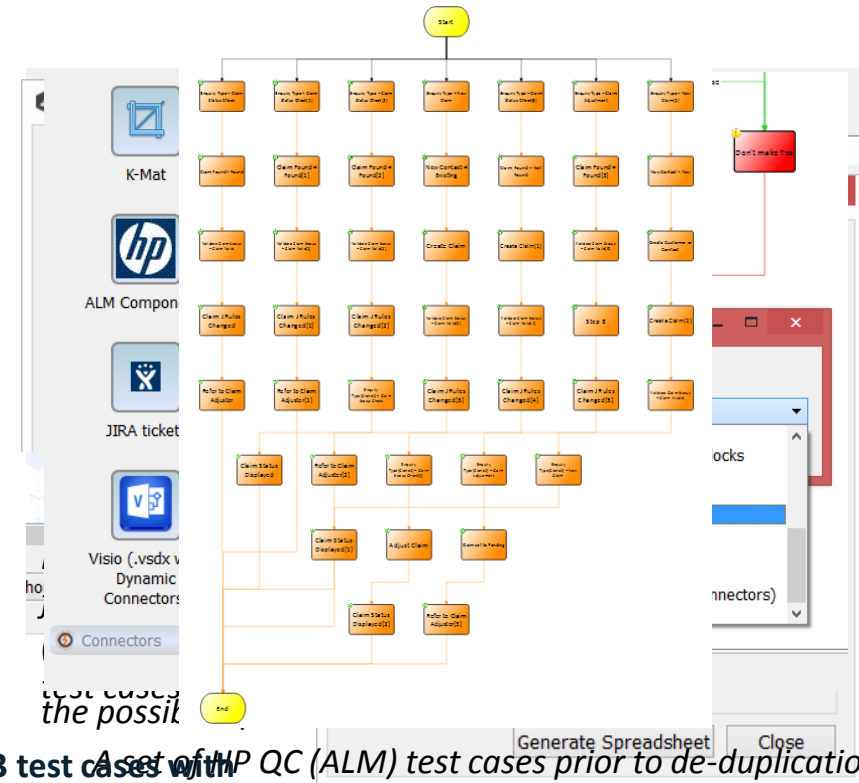


*With the click of a button, the test cases are repaired*

***At a large North American bank, it took 7.5 hours to manually update test cases when a change was made to a single ETL transformation. Agile Designer brought this down to just 2 minutes.***

# Improve existing test cases

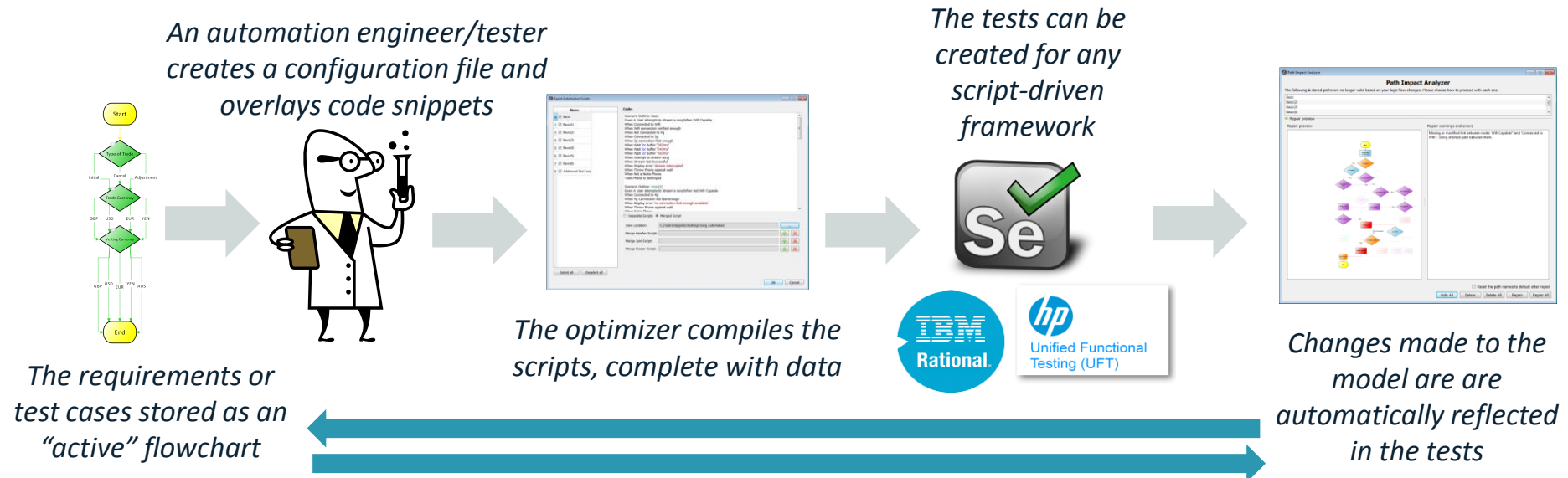
- **Import** existing test cases
- **Optimize** the test cases and test **more functionality** in **fewer tests**
- **De-duplicate** the test cases and test **more functionality** in **fewer tests**
- Push test cases back out to **existing tooling**



At a financial services company, one team relied on 3 test cases with just 5% coverage. CA Test Case Optimizer generated 12 test cases with 100% coverage in 3 minutes.



# Reactive Test Automation

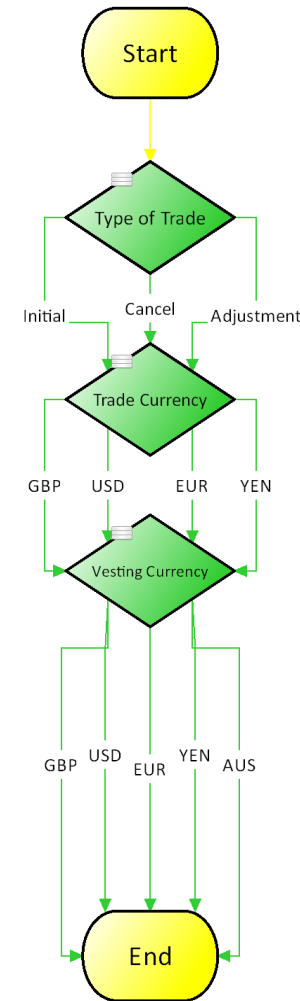


## No more manual, linear scripting, and all the benefits of the optimizer:

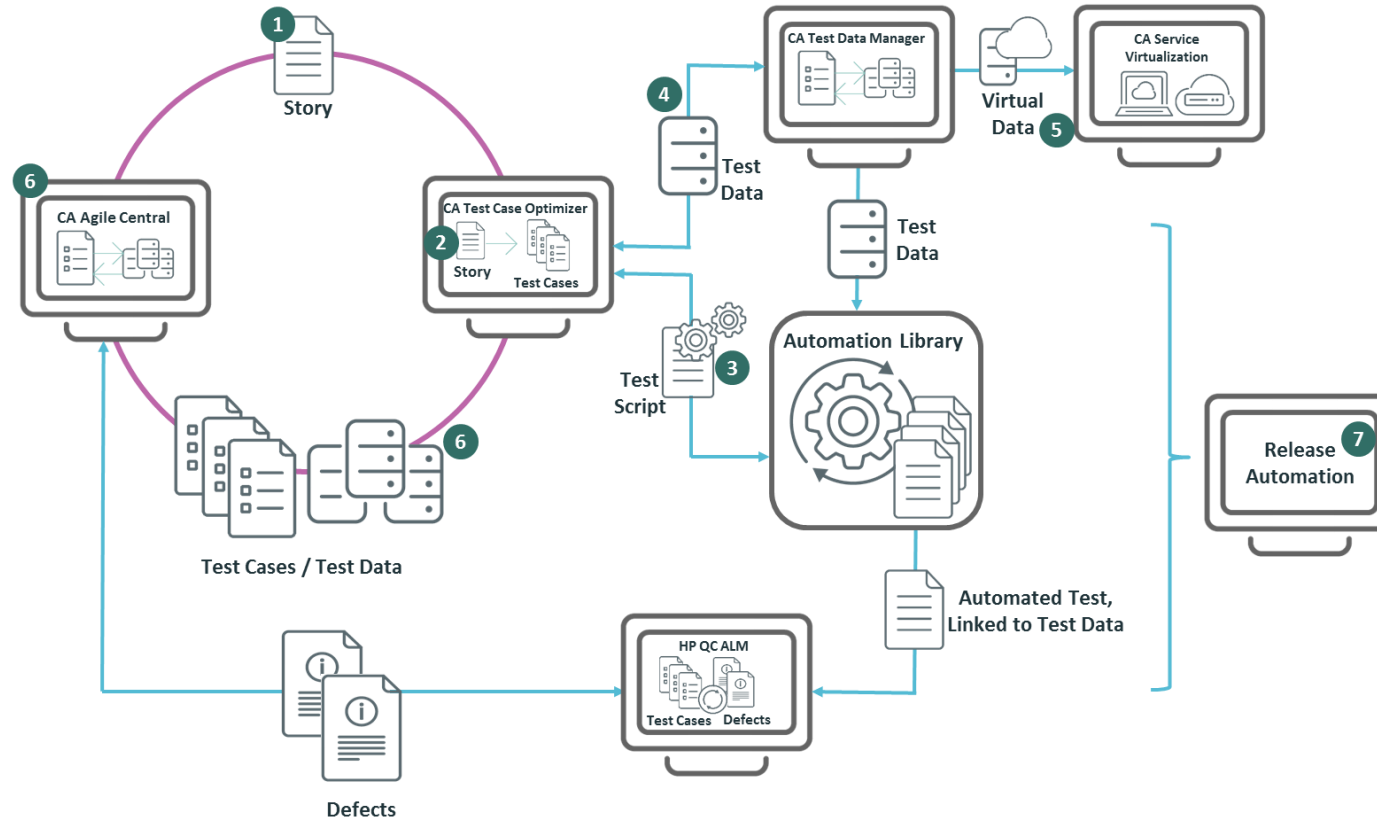
- Systematically and automatically derive optimized tests
- Maximum test coverage with no over-testing
- "One input, many outputs" – create and maintain test data, virtual end points and expected results as a part of automation
- React to change – auto-update the tests & create a re-usable library of test components

# Agile Requirements Designer

- Graphs sophisticated business processes and path modeling.
- Defines and optimizes requirements and test case designs
- Designs clear requirements, test cases, and even data and scripts, in one flow charting tool
- Significantly improves the quality of requirements, test cases and automated tests.
- Reduces costs of testing by detecting and creating the smallest number of test cases and scripts that have the maximum coverage
- Shortens test cycles and reduces the cost of testing without compromising the quality of the software



# Example: Using CA Agile Requirements Designer, Test Data Manager, Agile Central and HP ALM



1. A User Story is imported from Rally, and converted into an unambiguous, "active" flowchart model.
2. CA Test Case Optimizer generates the smallest set of tests needed for 100% coverage directly from the Story.
3. The test cases are converted and pushed out to automation frameworks.
4. All the data needed to execute the tests is found or created using CA Test Data Manager, and is "matched" to the tests themselves.
5. Meanwhile, CA Test Data Manager provides CA Service Virtualization with the virtual data needed to create any unavailable test environments. The virtual end-points can be defined in the User Story and Test Cases themselves in CA Test Case Optimizer.
6. Test Cases are pushed out to Rally and are integrated, along with defects, directly into HP QC ALM.
7. CA Release Automation provides automated deployment of the fully tested software.

# Why is This Important?

- Learn to determine inaccurate requirements/faulty apps
- Improve testing by:
  - Reducing over testing
  - Improved test coverage
- Improve software quality
- Reduce skyrocketing costs/defects

# Session Summary

- This session presented with the following information
  - What ARD can do to improve software quality
  - The structure of an ARD test case
  - How to create an ARD test case

# Questions?

# Comments?



# In Closing

## Questions after the webcast?

- Post them on the community site
- Contact me directly at [walter.Guerrero@ca.com](mailto:walter.Guerrero@ca.com)

## Upcoming TDM/ARD Community webcasts

- August 15<sup>th</sup>, 11:00 AM EDT, Harnessing the power of model based testing, James Walker and Joshua Taylor



## **Walter Guerrero**

Eng. Svcs Architect

Walter.Guerrero@ca.com



**in**