

# Overcoming Software Quality Challenges With Automated Testing

Travel & Hospitality Customer Success Stories



# Table of Contents

**03 Alaska Airlines Solves the Test Environment Dilemma**

**11 Caesars Entertainment Defines and Measures ROI for Test Automation**

**19 Sabre Virtualizes Web Services to Validate API and Data Interaction**



CASE STUDY

# Testing the Untestable

Alaska Airlines Solves the Test Environment Dilemma

## OVERVIEW

Alaska Airlines is primarily a West Coast carrier that services the states of Alaska and Hawaii with mid-continent and destinations in Canada and Mexico. Alaska Airlines received J.D. Powers' "Highest in Customer Satisfaction Among Traditional Carriers" recognition for twelve years in a row even recently winning first in all but one of the seven categories.

A large part of the credit belongs to their software testing team. Their industry-leading, proactive approach to disrupting the traditional software testing process ensures that testers can test faster, earlier, and more completely.

Learn how Ryan Papineau and his team used advanced automation in concert with service virtualization to rigorously test their complex flight operations manager software. The result: operations that run smoothly—even if they encounter a snowstorm in July.

RELIABLE &  
REPEATABLE TESTS

**100%**

ON-DEMAND  
AUTOMATED TEST CASES

**500**

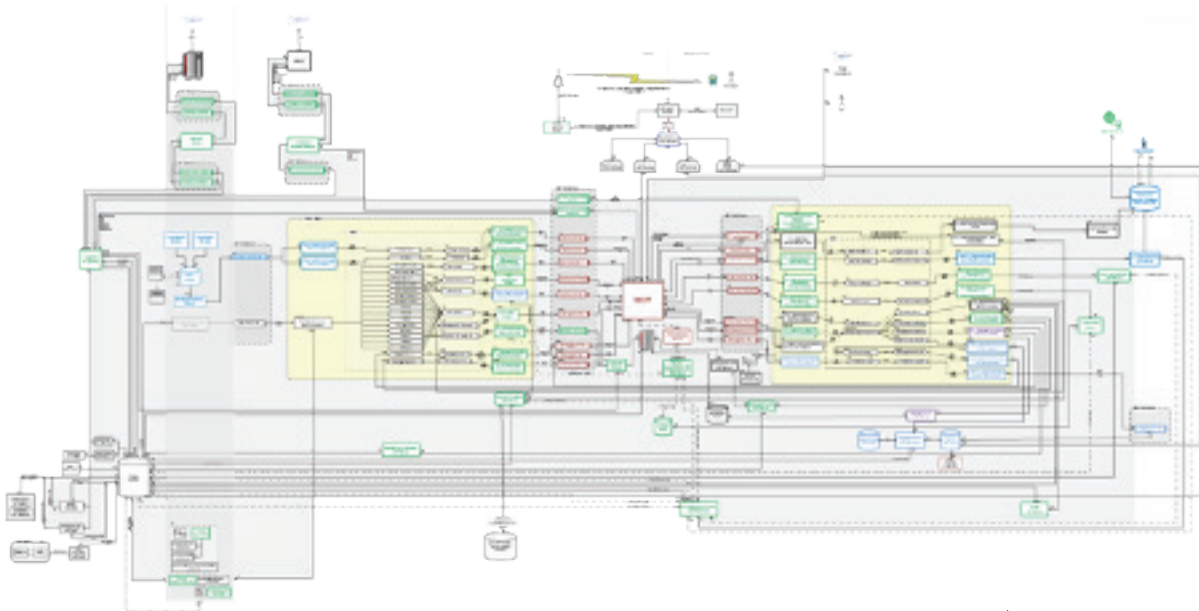
FALSE  
POSITIVES

**ELIMINATED**

## THE CHALLENGES

At Alaska Airlines, the flight operations manager software is ultimately responsible for transporting 46 million customers to 115 global destinations via approximately 440,000 flights per year, safely and efficiently. This software coordinates a highly complex set of inputs from systems around the organization to ensure flights are on time while evaluating and managing fuel, cargo, baggage, and passenger requirements.

In addition to the previously mentioned requirements, the system considers many factors including weather, aircraft characteristics, market, and fuel costs. It's highly complex with numerous dependencies (see Figure 1) and functions as the heart of a well-run airline.



### COMPLEX DEPENDENCIES & EVER-CHANGING TEST ENVIRONMENTS

The complexity and huge number of essential services as dependencies made it nearly impossible for Alaska Airlines to build repeatable tests. In addition, the test environment was shared with other teams and constantly changing. The changes impacted the flight management software and required persistent tracking and debugging. The ongoing resolution of these issues hindered their own development progress.

### MULTIPLE DATABASES UPDATING AT DIFFERENT FREQUENCIES

Data management was another large issue. It entailed various databases throughout the system updating at different intervals, making test data management a huge challenge. The various services in the system were another key challenge. It was nearly impossible to simulate real world events in a running production system.

Figure 1: Alaska Airlines Production Environment Architecture

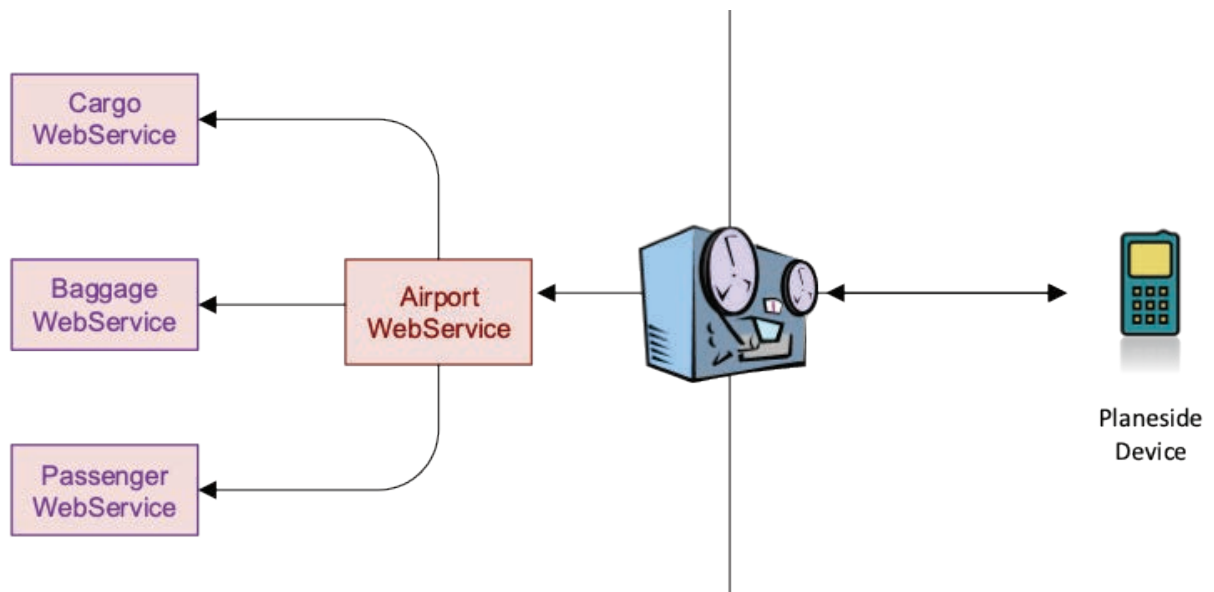
## THE APPROACH

Given the need to decouple their application under test from a very complex and constantly evolving environment, Alaska Airlines looked to service virtualization as a solution to their testing woes.

### REPEATABLE, CONCISE, & PRECISE TESTS

They use Parasoft Virtualize to decouple their application under test from the myriad of dependencies in their complex test environment. Alaska Airlines started using service virtualization as a proxy between their client devices and the airport web services. They ran through various scenarios to capture the requests and responses.

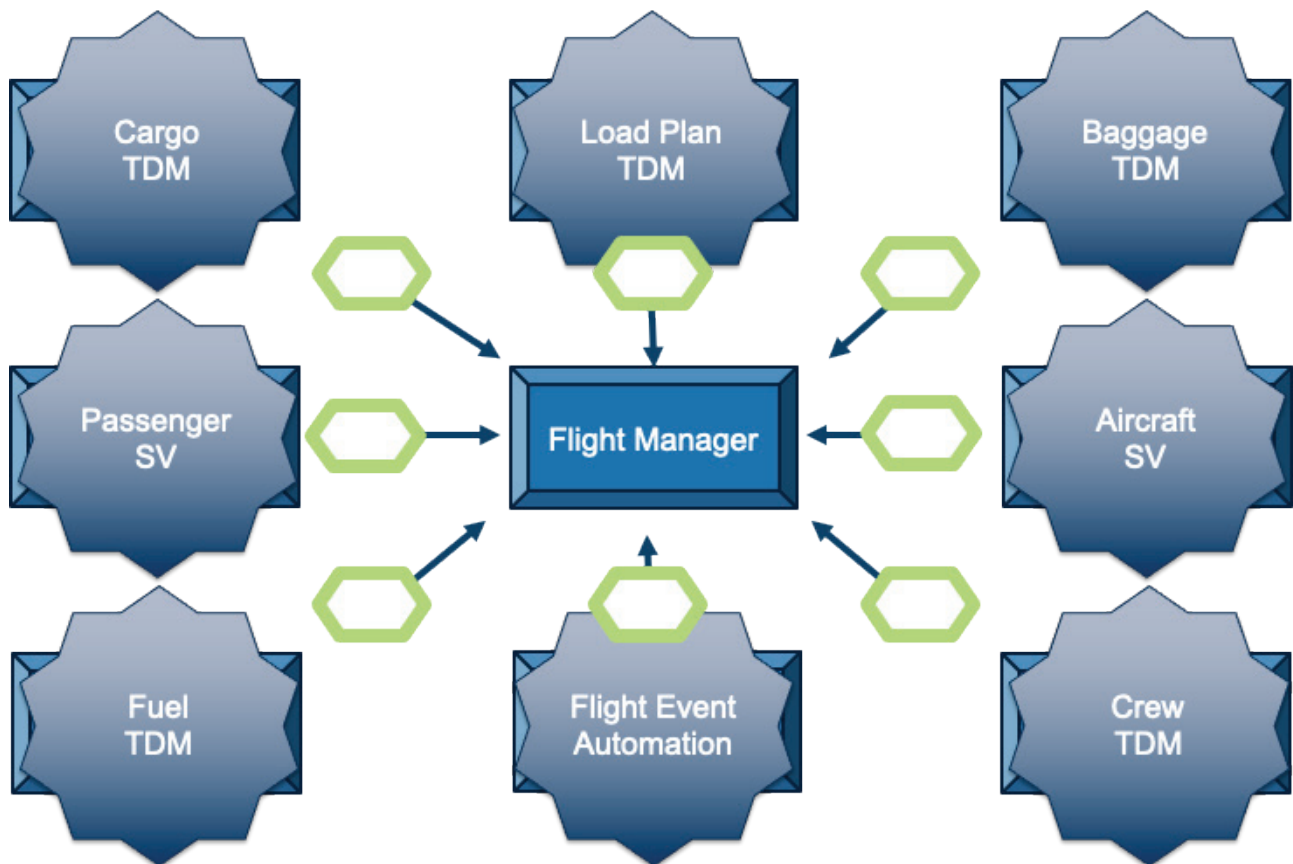
Figure 2: A simplified view of Alaska Airlines' virtualized test environment. In reality, the dependencies were numerous beyond the airport webservice.



These captured scenarios became the basis of repeatable, concise, and precise tests for later use. However, this was just one aspect of their system operation.

## VIRTUALIZED ENVIRONMENTS FOR EACH KEY SYSTEM

The full application isolation required more virtualization and test data management. Alaska Airline's virtualization environment included replacing all key systems with their virtualized equivalents as shown in the Figure 3.



Test data management was key to solving the issue with inconsistent databases in their real-world environment. Alaska Airlines decoupled the application from the dependent real-world databases by using Parasoft Virtualize test data management capabilities. They used IBM Optim in conjunction with custom T-SQL data transforms against data captured using Parasoft Virtualize.

Alaska Airlines captured enough data for a full day of real operations. Then deployed it within the test environment to create a "Groundhog Day" set of scenarios – repeating the same events and responses for a complete 24 hour cycle, progressing the time while keeping the scenarios consistent.

Figure 3: Alaska Airlines subsystem view. Each of these major systems were virtualized to decouple the flight manager for testing.



### ADVANCED SIMULATIONS FOR ROBUST TESTING

Alaska Airlines' solution for their passenger services is a point of pride. Using service virtualization, they were able to simulate passengers going through the complete lifecycle – booking, checking in, boarding – for all types of flights in their system.

The pilot's interaction with dispatch and fuelers is only a couple of the interactions they have with the system flight management system. It's nearly impossible to coordinate these interactions in real time without impacting production resources and personnel.

The team simulated these complex interactions with dispatch and fuelers. Getting consistent results, and no longer needed to leverage critical personnel to operate the airline.

### PARALLEL TEST ENVIRONMENTS TO SAVE TIME

Alaska Airlines now had the capability to decouple their flight management system from the real-world dependencies and a better ability to test it. One issue they faced was that running through their captured scenarios required an entire 24 hour period.

Testers needing specific scenarios had to wait until their set of events occurred, leading to hours of delay time. To solve this, Alaska Airlines operated seven virtualized test environments in parallel. Each one had its configuration customized to the team, using it in both performance and data feeds. The environments supported:

- » Development
- » Manual testing
- » Automated testing
- » Performance
- » Training
- » Business
- » Business rules testing



## RESULTS

Alaska Airlines freed themselves from the confines of testing in a complex and volatile real-world environment – to test the untestable. Once their flight management system test environment was decoupled from its many dependencies, they reaped the benefits of service virtualization.

### INCREASED TEST CAPABILITY

After adoption was complete, they became sophisticated users. Alaska Airlines increased their test capability to include cloning test data, fabricating scenarios, and creating tests where scenarios ran continually, vastly scaling the testability of their application under test.



*“If you can dream it, you can build it. We essentially went from being untestable to testable to now having a thriving demand for more of these capabilities.”*

—Ryan Papineau, senior software engineer at Alaska Airlines

“If you can dream it, you can build it. We essentially went from being untestable to testable to now having a thriving demand for more of these capabilities,” Ryan Papineau, senior software engineer at Alaska Airlines, explained.

### ELIMINATED FALSE POSITIVES & WASTED TIME

They have 500 automated test cases that leverage this virtualized test environment. Those test cases run as needed, improving their automated execution by magnitudes over conventional wait-and-see or hunt-and-peck testing models.

Thanks to all this simulation and automation, testers can test whatever scenario they want, whenever they want. Their tests are now 100% reliable and repeatable. They eliminated all the variables that previously caused false positives and wasted time.

Enabling testers to instantly access the exact test environment configurations that their test plans call for helps Alaska Airlines ensure smooth flight operations when faced with anything from a brief air traffic control outage to the improbable July snowstorm.



***Thanks to all this simulation and automation, testers can test whatever scenario they want, whenever they want. Their tests are now 100% reliable and repeatable.***

## SOLUTION BENEFITS

Another key learning experience and benefit for Alaska Airlines was a cultural change that made the organization more open to try new things. Now, the bigger the problem to solve, the easier the adoption of new techniques and tools is for the organization.

The benefits that the Parasoft service virtualization solution offered their team include the following.

**Decoupling from the complex production environment**, which allowed Alaska Airlines to isolate their application and generate the test cases and environment that would accelerate their testing.

**Repeatable, concise, precise test scenarios** were invaluable to Alaska Airlines and enabled them to scale the types and frequency of their testing. The team expanded and customized scenarios as needed and injected specific scenarios for special test cases. The repeatability of these tests sped up the development cycles and the consistency removed a serious challenge they had with the real-world operation environment.

**Parallel testing environments** reduced the wait time impact. The ability to clone and customize scenarios allowed Alaska Airlines to speed up their simulated test cycle. Parallelization sped up testing but also meant specific time-dependent scenarios occurred in shorter time frames.

**Test data management** enabled the team to decouple production databases and ensure data consistency, which plagued Alaska Airlines' operating environment. They reused test data captured from real-world systems to create life-like test scenarios that were difficult to coordinate in real time. Parasoft Virtualize test data management allowed the team to replicate databases consistently and in minutes. Something difficult to achieve in their real environment.

## TAKE THE NEXT STEP

Find the ideal service virtualization starting point for your team and identify the best deployment model for your requirements.

[Download the whitepaper.](#)



**CASE STUDY**

# Caesars Entertainment Defines and Measures ROI for Test Automation

## OVERVIEW

Caesars Entertainment is a global leader in gaming and hospitality. After merging with Eldorado Resorts, the company is the largest casino operator in the United States and includes 24 brands.

Caesars' top priority is its guests. They focus on building loyalty and value through a unique combination of great service, superb products, operational excellence, and technology leadership.

In an endeavor to modernize and expand their customer-focused loyalty program, Caesars chose to integrate Salesforce as the foundation for the systems. To ensure successful implementation, they couldn't afford to gamble with quality.

With test automation a critical factor in delivering a high-quality customer experience, Roosevelt Washington, senior IT manager of quality assurance at Caesars Entertainment, took the lead to successfully adopt test automation practices and deliver measurable value to the business.

SAVED IN  
ONE YEAR

**>\$1 million**

REDUCED TEST  
EXECUTION TIME

**97%**

IMPROVED UI  
TEST AUTOMATION

**>96%**



## THE CHALLENGES

As Caesars has grown through acquisitions over the years, so has the number of developed applications. The result is multiple disconnected systems across multiple companies.

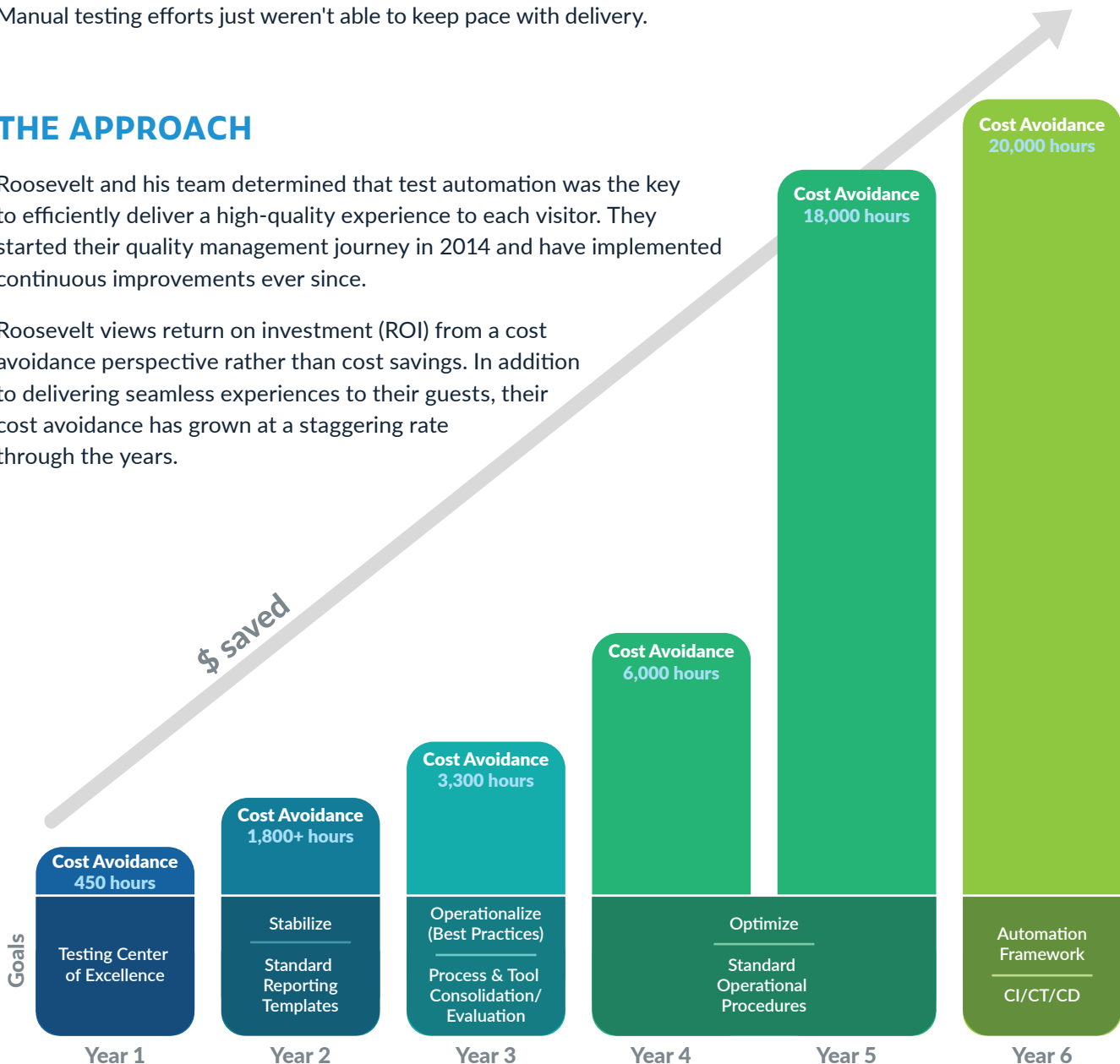
It's extremely important to Caesars to create a seamless experience for their guests. That means that no matter which of the acquired 22 new properties guests choose to visit, they have a consistent experience. For example, they can take their reward card to any slot machine on any property and it will work the same way. It's a smooth, flawless interaction.

There's a great deal of integration work and testing required within a complex system to ensure guests have a common and consistent experience. Manual testing efforts just weren't able to keep pace with delivery.

## THE APPROACH

Roosevelt and his team determined that test automation was the key to efficiently deliver a high-quality experience to each visitor. They started their quality management journey in 2014 and have implemented continuous improvements ever since.

Roosevelt views return on investment (ROI) from a cost avoidance perspective rather than cost savings. In addition to delivering seamless experiences to their guests, their cost avoidance has grown at a staggering rate through the years.



### BUILD A TESTING CENTER OF EXCELLENCE

With the decision to create a testing center of excellence came the start of a development roadmap. Committed to improving quality and testing standards and practices, the QA team took an incremental approach beginning with the staff. The first steps on Caesars' journey were to define and implement test automation and quality management best practices.

**Cost Avoidance: 450 hours**

### STANDARDIZE REPORTING TEMPLATES

In 2015, Caesars took the next steps to improve their quality assurance (QA) processes:

- » Stabilized testing practices.
- » Standardized reports to best communicate test and product status.

**Cost Avoidance: 1,800+ hours**

### CONSOLIDATE PROCESSES & TOOLS

The following year, the QA team evaluated and consolidated all of their existing testing tools, opting for the best of market from their point of view. They worked to standardize processes and tools across the entire enterprise.

**Cost Avoidance: 3,300 hours**

### OPTIMIZE A STANDARD OPERATING PROCEDURE

During 2017 and 2018, the QA team worked to optimize their test frameworks and standardized operating procedures.

**Cost Avoidance: 2017: 6,000 hours**  
**2018: 18,000 hours**

### CREATE AN AUTOMATION FRAMEWORK

For 2019, Roosevelt and his team moved to continuous integration/continuous deployment (CI/CD) pipelines. They created an automation framework, which is two-fold:

- » Frontend
- » Backend

The frontend is like self help. Super users—even those without coding knowledge—can select use cases from an interface to run automatically.

The backend is where environment simulation is set up for continuous integration, continuous testing, and continuous delivery.

**Cost Avoidance: 20,000+ hours**

## CONTINUOUS IMPROVEMENTS

2020 brought COVID-19, which disrupted the hospitality industry and many of Caesars' initiatives, including engaging in service virtualization. That plan will be carried over to 2021 and beyond when they will incorporate web UI testing using key technologies and deploy service virtualization to improve their API test strategy.



## KEY CONSIDERATIONS

When implementing software testing automation, there are essential factors to consider.

- » Gain the support of leadership.
- » Anticipate a change in culture. Just like any change, automation can be scary, but it ultimately empowers development and QA teams, and benefits the entire organization.
- » Quantify your ROI and communicate it to the key stakeholders.
- » Get strong alignment with resources and access to tools. In some instances, you may not have the bandwidth necessary from an internal perspective to reach an initiative.
- » Allot time for script maintenance.

Caesars' quality assurance team shared a couple of experiences that put some of the key considerations into perspective.

### CHOOSE A STRONG PRODUCT & A STRONG VENDOR

The team had a product they wanted to get into service virtualization. After developing the use case and providing it to a vendor, the virtualization failed.

That particular vendor wanted to charge Caesars to tell them why their product was failing. Roosevelt looked for other vendors and discovered Parasoft. After running the same use case, the same problem occurred.

"The difference? Within 30 minutes, Parasoft's development team had a hot fix and we were on our way," explained Roosevelt. "We had a successful proof of concept for service virtualization. You need a strong product and a vendor with strong support."

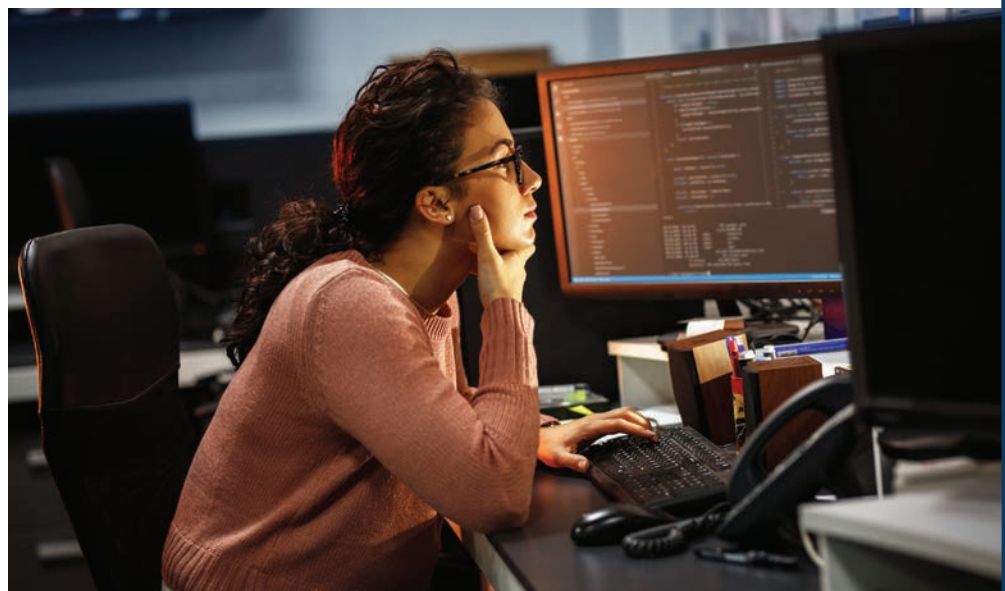
*"Within 30 minutes, Parasoft's development team had a hot fix and we were on our way. We had a successful proof of concept for service virtualization."*

—Roosevelt Washington, Senior IT Manager of QA

### SAVE TIME REPURPOSING SCRIPTS

Caesars saves time with performance testing by using the same scripts with minimal changes for load testing. Roosevelt says, "You don't need to have yet another product to do that, and that's what I love about using Parasoft SOAtest for API testing."

*Caesars saves time with performance testing by using the same scripts with minimal changes for load testing.*





## THE RESULTS

Caesars is walking the path of continuous improvement, always examining what they're doing and how to make it better. To communicate the ROI to key stakeholders within Caesars, Roosevelt came up with an automation ROI formula that shows year-over-year cost avoidance using test automation practices.

### Automation ROI Formula

$ROI = ((\text{Manual Time} - \text{Automation Run Time}) \times \text{Resource Cost}) - (\text{Cost to Deliver} + \text{Monthly/Annually})$

Attaching numbers, below are examples of how the formula works.

### One Automation Regression Suite Calculation

With  $M = 31$  hr.,  $A = 1$  hr., and  $R = \$50$ , the calculation is  $(31 - 1) * 50 = \$1,500$  - Deliver Cost (5 hrs. x  $R = \$250$ ), which results in a total cost avoidance of  $\$1,250$ .

The cost to deliver includes products that you're using, burden rates, and similar items. Based on the formula, Caesars saved  $\$1,250$  running one regression suite just one time.

### Annual Automation Calculation

$AA = (\$1,250 \times \text{Run Frequency: 52 Weeks}) = \$65,000 - (\text{Deliver Cost: SW/HW/BR, Monthly or Annually})$

### Annual Automation Savings

When you look at annual automation, take  **$\$1,250$  savings** and multiply it by the frequency in which you run tests. In this scenario, it's one per week for **52 weeks**. That's  $\$65,000$  year-over-year minus the cost to deliver software, hardware, and your burden rate, whether you calculate it monthly or annually.

## DEMONSTRATING ROI TO LEADERSHIP

The calculations based on the automation ROI formula are what Roosevelt presents to leadership at Caesars. It demonstrates a measurable reason to automate testing: the savings in costs you avoid (cost avoidance) outweighs the cost to purchase products in the hardware and software testing tools.



**When you multiply those 20,000 hours of automated testing by \$50 per hour for the cost of a resource to do it manually, the cost avoidance is well over a million dollars.**

**Improved UI test automation by more than 96%.**

**Reduced API test execution time by 97%.**

#### **AVOIDING COSTS**

Caesars' testing and ROI automation endeavor started in 2014 with 450 hours of cost avoidance. Each year their annual cost avoidance hours increase exponentially. In 2019, Caesars' cost avoidance based on automated regression testing was 20,000+ hours.

When you multiply those 20,000 hours of automated testing by \$50 per hour for the cost of a resource to do it manually, the cost avoidance is well over a million dollars.

#### **SOLUTION BENEFITS**

By building a scalable and maintainable test automation strategy with the Parasoft Salesforce testing solution, Caesars accelerated testing and delivery. Here are two of the main factors contributing to this acceleration and cost avoidance:

- » **Improved UI test automation by more than 96%.** Executing UI tests took excessively long, taking many days, before automating testing with the Parasoft solution.
- » **Reduced API test execution time by 97%.** Using key technologies, they were able to test UI-driven APIs for stable and rapid feedback versus browser-driven scenarios.

## **TAKE THE NEXT STEP**

Find out how to choose the right API testing solution for your organization. [Download the whitepaper.](#)



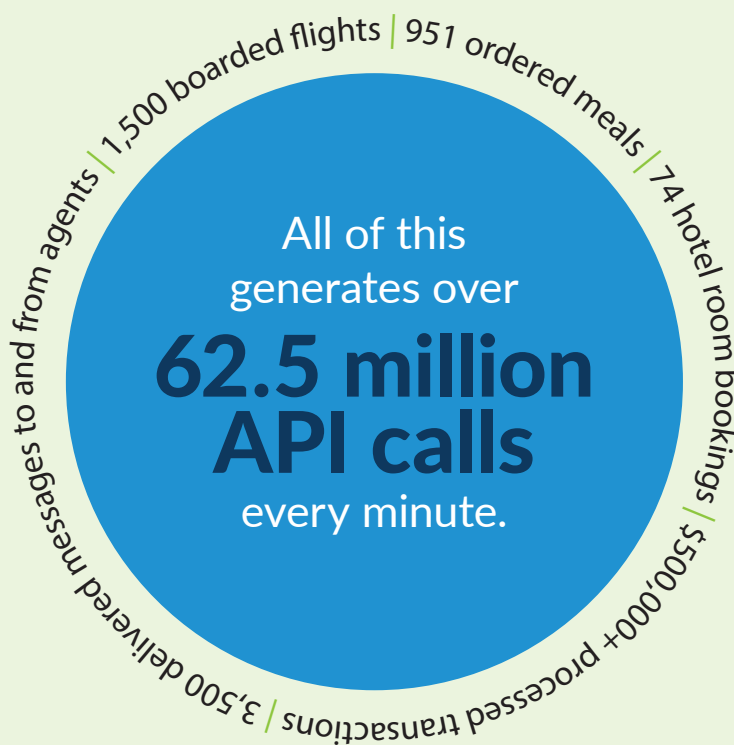
**CASE STUDY**

# Virtualize Web Services to Validate API and Data Interactions

## OVERVIEW

Sabre is the world's largest provider of airline and hotel technology, used by more than a billion people around the world. Their systems must work 24 hours a day, every day, processing more than 60,000 transactions per second at peak demand. Sabre customers include airlines, airports, cruise lines, government, hotels, online travel agencies, railway carriers, and more.

Sabre hosts a global distribution system (GDS) with massive amounts of data. A typical minute inside Sabre's systems includes:



REDUCED TIME  
& EFFORT

**67%**

ANNUAL  
SAVINGS

**\$720,000**

PERFORMANCE  
ISSUES & RISK

**ELIMINATED**



## THE CHALLENGE

To transition Sabre's core software systems from its historic proprietary base to services exposed as APIs, the company encountered new challenges. Its software systems require high reliability and availability, and the new APIs would need to meet those standards.

Sabre faced big challenges.

- » Dependencies on a diverse set of highly complex downline systems from various partners such as airlines, cruise lines, and railway carriers.
- » The sheer number of protocols and test cases needed to ensure quality, reliability, and good performance.

Performance of these downline systems impacted the pace of testing because they were live systems with real-world delays. In addition, the total complexity and number of different downline systems meant there were numerous service descriptions and protocols to satisfy. All this made testing difficult.

They decided automated testing solutions were needed to handle the workload. However, using their own solutions, they discovered that simple mocking (substituting real systems with simple request-response stubs) was inadequate. As a result, Sabre investigated Parasoft solutions for their complex test environment.

## THE APPROACH

To reduce operational costs, improve flexibility to meet changing business requirements, and provide simplified automated access to its travel services, Sabre found it needed to move from the historic proprietary architecture its systems were based on and migrate to an open standards-based architecture.



## API TESTING

The move to APIs presented new challenges to Sabre's software testing infrastructure. Interaction with third-party services would require the means to test interoperability between those services and appropriately simulate often-complex transactions across the multiple services, representative of the business processes that would be transacted by them.

In production, the lack of availability of a service at any time could result in significant losses to important companies in the transportation industry. The procedures the company initiated to test software quality and reliability had to be thorough and 100% accurate.

As Sabre would be developing and deploying many services on an on-going basis, it needed a repeatable testing methodology that promoted system stability through continuous integration. Additionally, the test methodology needed to establish a baseline for certification of production readiness that each service would need to meet before qualifying for production deployment.

The company defined three primary goals to achieve in order to deliver quality services:

1. Automated test case generation and execution
2. Repeatable regression testing
3. Breadth of code coverage

Sabre uses Parasoft SOAtest to implement an effective testing and certification solution for their automated web services testing process. Sabre's business analysts leverage SOAtest to quickly create more thorough test cases during development to verify that the service meets requirements.

Then the QA team uses SOAtest to execute these test cases along with a standard regression suite of tests to validate and certify that the service meets reliability and performance objectives. This process has allowed Sabre to prevent serious defects from reaching production and has reduced the time and effort to certify a new service by two-thirds. This productivity gain translates into a savings of \$720,000 annually through automated testing.

***"The company will save \$720,000 annually through the automated testing provided by Parasoft solutions."***

## VIRTUALIZATION

Sabre investigated possible solutions in the market and started a proof-of-concept to use Parasoft Virtualize for their test environment challenges. The evaluation criteria considered whether the solution could handle the needs of such a complex system, the high volume of API calls, and different kinds of client applications. They also assessed how well Parasoft Virtualize would work for testers and developers and how friendly it was to non-technical users.

After a successful proof-of-concept, Sabre integrated the Parasoft solution into their testing environment. Initially, Parasoft Virtualize was placed between their application under test and the downline systems. In this case, the tool captured all the API traffic passing from the application under test to the rest of the system. This API capture was beneficial in recording all the requests and responses coming to and from the downline system. However, they needed more than simple record and play back functionalities.

Using the powerful machine learning capabilities of Parasoft Virtualize, Sabre used the automatically generated test scenarios built from heuristics that group related API calls together. Pleased with the results, they fine-tuned these scenarios into reusable, repeatable test cases.



***Sabre was able to automatically generate thousands of test cases based on traffic analyzed from diverse third-party application clients.***

In the end, Sabre was able to automatically generate thousands of test cases based on traffic analyzed from diverse third-party application clients such as airline and hotel purchase systems. The team used their service descriptions to generate mocks before those services were fully developed.

Once the test cases coalesced from the recorded traffic, Sabre needed a test data management solution for these large test suites. Sabre leveraged Parasoft Virtualize to mask the data to obscure private information and stored the test data sets in a database. These data sets formed the basis of their dynamic test scenarios.

***The team values how easily Parasoft Virtualize allows testers to switch between live and virtualized systems as needed.***

## THE RESULTS

With Parasoft SOAtest, Sabre effectively transitioned its software systems to APIs that now process over five million transactions daily. By automating their API testing, Sabre achieved the test coverage and repeatability necessary to ensure that its services consistently meet its high reliability and availability standards.

Using Parasoft Virtualize to simulate Sabre's complex downline systems and a flexible suite of automated test cases means that Sabre can now decouple their test environment from the real system. In fact, the team values how easily Parasoft Virtualize allows testers to switch between live and virtualized systems as needed. Sabre eliminated the performance issues and risk of using real downline systems while greatly accelerating their testing cycles.

***Sabre eliminated the performance issues and risk of using real downline systems while greatly accelerating their testing cycles.***

Integrated into the Sabre development lifecycle, Parasoft test automation solutions allow Sabre to deliver a richer set of capabilities to its customers. At the same time, they can ensure that the release of every service is stable and meets business requirements before it goes into production.







## SOLUTION BENEFITS

Parasoft helped Sabre achieve their application and API testing goals with integrated automated testing tools that significantly reduce the time needed to deploy services and ensure these services meet reliability, availability, and functional requirements.

Automating their test environment and decoupling it from a complex, real-world production system provided several benefits.

**Automated API test case generation** enabled Sabre to generate more thorough test suites for requirements verification, reducing the creation time by one-third.

**Repeatable, dependable, and simulated services** allowed Sabre to create repeatable API test scenarios with partner-specific test data sets to test their applications in each partner ecosystem.

**Virtualizing the production systems** that may be unavailable or performing poorly for testing reduced developer and tester down time.

**One workflow with a highly available, integrated test environment** allowed Sabre to streamline their development and testing pipeline.

**Service virtualization with dynamic and relevant service responses** enabled Sabre to create a realistic virtualized test environment. It also allowed the team to perform specific tests with injected data as needed.

## TAKE THE NEXT STEP

Find out how to choose the right service virtualization solution for your organization. [Download the whitepaper.](#)

### ABOUT PARASOFT

[Parasoft](#) helps organizations continuously deliver quality software with its market-proven, integrated suite of automated software testing tools. Supporting the embedded, enterprise, and IoT markets, Parasoft's technologies reduce the time, effort, and cost of delivering secure, reliable, and compliant software by integrating everything from deep code analysis and unit testing to web UI and API testing, plus service virtualization and complete code coverage, into the delivery pipeline. Bringing all this together, Parasoft's award winning reporting and analytics dashboard delivers a centralized view of quality enabling organizations to deliver with confidence and succeed in today's most strategic ecosystems and development initiatives – security, safety-critical, Agile, DevOps, and continuous testing.