



Enterprise Test, Workflow, and Infrastructure Management Automation

Executive Summary

The Testing Community faces numerous challenges as software intensive, complex, and integrated enterprise system of systems testing becomes the norm and not the exception. While Federal and Civilian test agencies (e.g., DoD, DoE, DISA, FAA, VA...) do an excellent job of single domain testing (e.g., developmental (DT), operational (OT), functional (FT), interoperability (IOP), and IV&V testing), conducting integrated and agile testing in a federated and virtualized environment with shared resources has become increasingly difficult. Today, testing is comprised of a combination of manual and semi-automated test processes using a wide variety of test tools and data collection techniques in a stove-piped environment, which does not efficiently use valuable resources. As test complexity increases, the need for integrated and automated data collection, scheduling, resource management, and unified test development rises exponentially. Large commercial organizations such as Cisco, Verizon, and Microsoft, whose bottom line depends on efficient testing of products, face similar challenges. These companies take an approach that standardizes and automates test procedures and then fully controls the test process through automated resource management.

Technical Systems Integrators, Inc. (TSI) has studied and delivered proven automation solutions to handle the challenges of today's testing and infrastructure environments. Given the success of our solutions and large Return on Investment experienced by our customers, we see our solution providing the greatest potential to automate and enhance commercial and government entities testing effectiveness and quality, while reducing test cycle times, costs and resources needed to deliver the desired goals. This capability is provided by TSI and incorporated into our AgileWARE® (Agile Workflow And Resource Enablement) solution using CloudShell™, the leading infrastructure automation software for service-oriented architecture systems, hardware, software and networks.

CloudShell™ is an end-to-end automation framework solution that encompasses the entire test environment and lifecycle of infrastructure and the workflows inherent. The system provides code-free test authoring, automatic test execution, advanced test management, complete lab and resource management, and comprehensive test reporting and analysis. Through its comprehensive automation of the testing process, CloudShell™ has assisted entities like Cisco, Juniper, Verizon, AT&T, DISA, PACOM, Bosch, and many others to improve product quality, shorten time to market, and significantly reduce capital and operational expenditures. Notable examples of reductions in test-cycle time with CloudShell™ include a reduction of 50+ days to 8 hours for DISA and 3 days to 1.5 hours for Telstra. AgileWARE® implemented by TSI can provide test organizations with the ability to better control and automate test configurations, enable enterprise-level control of test assets, allow automated regression testing, provide real-time test metrics, improve the scheduling and coordination of testing, share test automation across the organization, and provide complete lifecycle management from development through production.

Contents

Introduction	2
Resource Management Challenges	2
The Requirement: Agile Testing and Management	4
The Solution: AgileWARE® – Agile Workflow And Resource Enablement - An	
Automated Test and Resource Management Framework	4
Automated Topology Configuration	5
Automated Reservation and Scheduling	5
Coordinated Execution	6
Enterprise Data Repository	6
Unified Test Development	6
Benefits	6
Configuration Control for Test Setups	6
Enterprise Asset Control of Resources	7
Higher Levels of Test Automation	7
Automated Regression Testing	8
Software-based Scheduling and Coordination of Testing	8
Automated Test Efficiency Metrics	8
Enterprise Level Solution	9
Summary	10
 ANNEX A: Candidate Implementation Plan	 A-1
ANNEX B: Return on Investment Results for Candidate Implementation Plan	B-2

Introduction

This white paper describes the challenges and potential solution for test initiatives, either internal or external to the consumer, to transform from a semi-automated and non-persistent test environment to a fully automated and federated capability that enables Testing as a Service (TAAS). Additionally, provided is a return on investment (ROI) analysis which shows that AgileWARE® implementation can significantly reduce operational and maintenance costs to allow for a recoup of initial investment in approximately 6 months or less.

AgileWARE® provides a code-free hardware interface editor and complete Python APIs for the quick development of drivers/shells. The application uses an intuitive, flowchart-like Graphical User Interface (GUI). The GUI assists test engineers to independently and easily build drivers or entire driver collections (libraries) that seamlessly interface with most brands and types of test equipment. The GUI also provides the interface with network test equipment, such as Automated Internet Protocol Traffic Generators, Layer 1 switches, application test tools, test automation frameworks and more. With CloudShell Authoring and Python driven test scripts, the numerous scripts and resources that are commonly required for equipment interfacing are now replaced with a standards based (TOSCA like) shells that is easily created and deployed. These shells provide for autoloading of assets, configuration management and complete control over infrastructure assets no matter where the infrastructure exists (Physical, Virtual, Cloud, Hybrid). This translates into test engineers learning how to interface with test equipment and Devices Under Test(DUTs) through one GUI rather than having to learn a separate management interface for each asset being controlled.

The following sections describe common challenges that organizations face in terms of resource management and test orchestration in environments that are either internally or geographically dispersed. The following sections also address the requirements for agile testing and propose a solution to these challenging problems.

Annex A provides a phased implementation plan of a candidate AgileWARE® installation and a detailed candidate-phased implementation plan that supports 5 test positions and 10 infrastructure consumers(including 1 manager) for a single enterprise group. Annex B provides the results of the ROI analysis conducted for such an implementation. This ROI is based on historical data derived from implementations of AgileWARE® capabilities by commercial entities.

Resource Management Challenges

The resource management challenges associated with current test environments include setup, utilization, sharing, controlling, and training.

Setup: As network environments and equipment become more complex, the time required for the configuration and setup of test equipment and the supporting infrastructure increases. Required configuration adjustments for different test batches or Devices Under Test (DUT)/Systems Under Test (SUT) dictate manual wire-up and re-cabling of equipment, sometimes leading to time-consuming configuration errors. Configuration management across

resources and all aspects of the test environment is challenging. Figure 1 shows the most common issues with a partially automated test strategy.

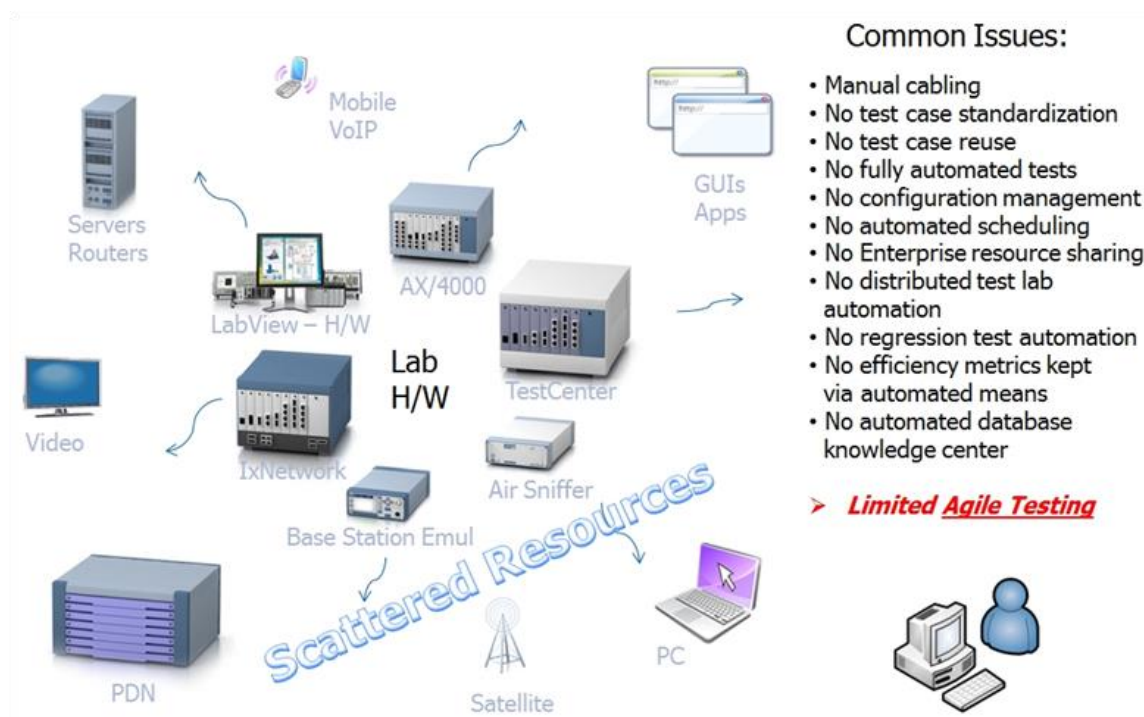


Figure 1. Common Problems of Semi-Automated Testing
Manual Processes Lead to Systemic Inefficiencies

Utilization: It is expensive to equip and maintain test laboratories. Physical resources such as traffic generators or switches can cost millions of dollars and are expensive to replace if they are damaged or become outdated. Logical resources, and managed functionalities like switch node configurations, software licenses, and maintenance fees are also often costly. Any level of utilization less than 100 percent constitutes a loss for the organization.

Sharing: In most organizations, test resources must serve multiple test teams. Sharing these resources can be quite frustrating. Ensuring that test equipment is available when required, and guaranteeing tests are not interrupted seconds before they end, is often an impossible task. Resource use conflicts between individual testers and infrastructure consumers results in inefficient use of expensive equipment and in longer, unpredictable testing time and effort.

Control: When sharing resources across large or distributed test labs, resources are often stored in test-equipment farms. Control over the equipment becomes increasingly difficult as more and more Layer 1 switches are required to automatically manage resource cabling and connections. Control gets even trickier as large clusters of switches with complex topologies are required.

Provisioning: Lab Assets require constant configuration control when being used for different test use cases requiring time and effort to reset and configure assets. This effort significantly slows test throughput when configurations are not properly set and reset between use cases.

Training: Test equipment and infrastructure applications are controlled by a myriad of languages and interfaces. Before test and infrastructure users are able to conduct testing and other activities, they must learn those languages and interfaces for specific items of test equipment. If engineers move on to other laboratories or leave, the expertise they acquired is lost.

The Requirement: Agile Testing and Management

The increasing support demands of new Device/System Under Test (DUT/SUT), new test equipment, and additional personnel required to test these items continually drive support costs in an upward spiral. Without the capability of sharing resources, automating tests, standardizing tests, reusing tests, automated save/restore of configurations and ability to make all aspects of the tests available upon request, the cost of manually performing these tasks becomes prohibitive.

The good news is that the test world is not alone in this requirement. As companies become more and more dependent on internal and external networks the need for updates to these networks and validation of these updates is critical.

The Solution: AgileWARE® – Agile Workflow And Resource Enablement - An Automated Test and Resource Management Framework

The solution that numerous companies are turning to incorporates sophisticated automated test capabilities that automatically calculate switch topology paths, can manage clusters of Layer 1 switches, asset/tool configurations and provide automated resource management that incorporates scheduling and reservation mechanisms. Figure 2 is a graphical overview of what an automated system would look like.

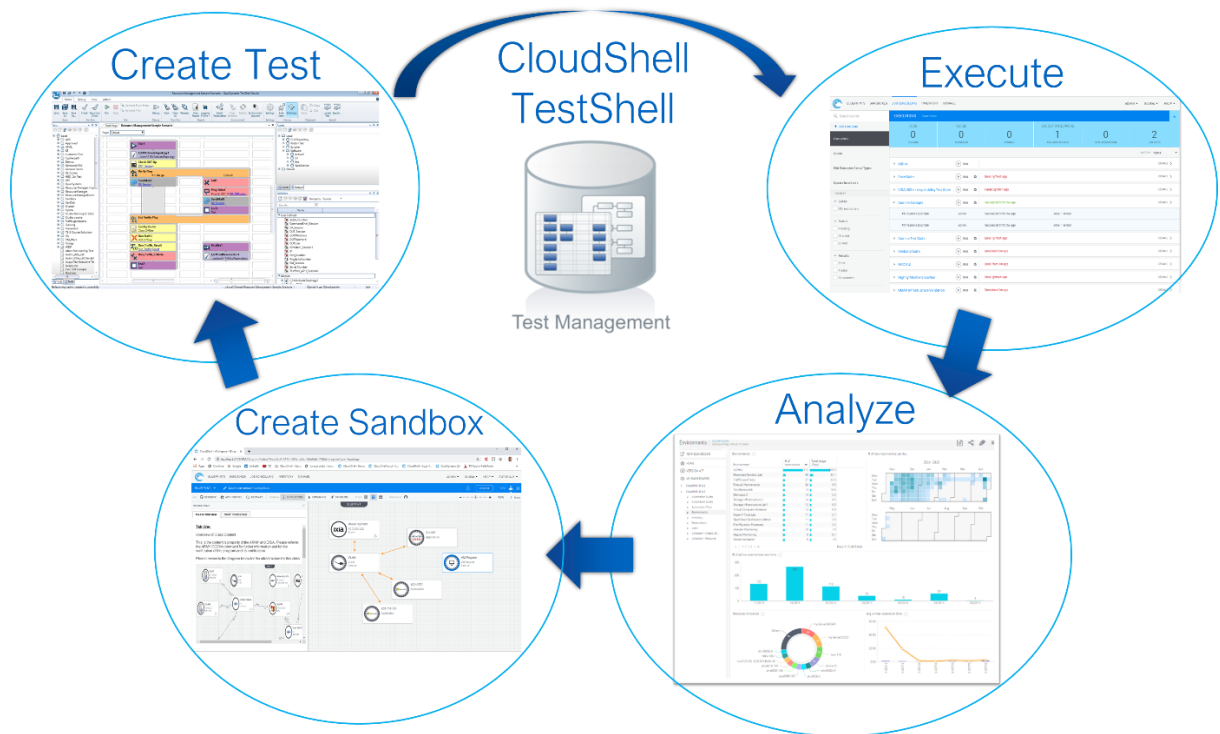


Figure 2. Enterprise-Level Automated Test Lab Management
AgileWARE® Capabilities can Significantly Enhance All Aspects of Automated Testing

Automated Topology Configuration

To economize with configuration and manual cabling efforts, Layer 1 switches can provide outstanding efficiencies. The Layer 1 switch provides a wire-once technology, which users can employ to connect any port within the system to any other port, simply with software commands. Once the lab network and equipment are connected, test topology changes are accomplished by a simple mouse click, and there is no further need for manual cabling.

As multiple labs share the same resources, multiple Layer 1 switches are necessary, and topologies become complicated. An automated system can wrap all the switches into one cloud or cluster and then control the cloud using a background service to automatically calculate the optimal topology according to test requirements and definitions. By using Abstract topology configurations, the system can figure out the optimal provisioning and setup without the user having to deal with these tasks saving large amounts of setup time and reducing the errors in provisioning tasks. The result is that testers and infrastructure users are blissfully ignorant of and unaffected by the switching solution, which is run for them automatically.

Automated Reservation and Scheduling

An automated service that can lock and release test resources according to predefined rules can avoid utilization conflicts between test events at multiple locations. With all resources under a centralized repository, the management of resource reservations between test events, test development teams, and other resource customers is optimized, thus guaranteeing optimal resource usage and minimal test interruption.

While an automated scheduling capability is paramount, there is a requirement for some flexibility in an automated system. For example, if a test overruns its allocated time and requires an additional few minutes to finish a long and complex test process, it would be wasteful not to allow it to run to completion by reallocating a resource to another test. To avoid such cases, and to ensure the test with the highest priority gets all the resources it requires, a rules engine that defines a hierarchy for tests and resource users, and that provides an automated conflicts management system is a great advantage.

Coordinated Execution

When dealing with growing multi-site and large-scale testing environments, achieving timely and cost-effective test execution requires not only well-managed resource sharing but also control over test stations, test requirements, and an ability to match and coordinate all these components in a synchronous manner. To automatically allocate tests to the appropriate available station, a centralized application is required that can dynamically abstract resources by defining their structure (chassis, card, port, etc.), follow protocols and rules, and remotely control all resources and stations. With these tools, a well-coordinated execution can be achieved.

Enterprise Data Repository

To support such a framework, fully scalable distributed database architecture is required to handle the control and sharing of test automation libraries and functions, user and group controls, and instrument resources. Proper control within a database framework simplifies the reporting of results from tests and provides management visibility into test results. Additional metrics such as test quality, test resource use, and overall test metrics for laboratories and test stations can be reported so that an overall evaluation of lab usage and performance can be conducted.

Unified Test Development

Test engineers should be able to focus on testing, not learning new languages and interfaces to control the test environment. An easy-to-use, drag-and-drop GUI provides for quick programming-free test sequencing and can support parallel execution and event-driven test scenarios so that non-programmers can develop complex and powerful tests. Built-in integration of script languages and software resources, combined with simple parsing and analysis rules, allow easy construction of elegant test scenarios as well as development of shareable functions and tests for reuse within and outside each lab. Control of the test resource topology from within the library of test functions allows full configuration management of test regressions and resources so that tests can be independent of the resources. Support for Python based test automation and robotic process automation (RPA) tools can be supported. This provides for greater flexibility because it frees testers from having to rewrite tests when topologies or test resources are changed or updated.

Benefits

Configuration Control for Test Setups

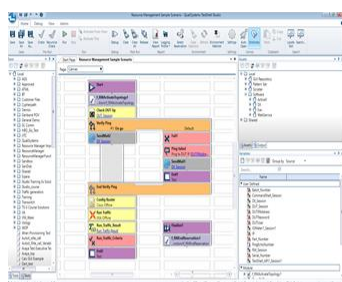
Test personnel will be able to control test topologies and configurations and save them away, saving tremendous amounts of time in configuring test resources to meet the needs of a particular test. Interruptions for dynamic reallocation and configuration changes can be

handled with complete save and restore functions across the entire sandbox topology allowing for rapid reconfiguration on demand. Personnel can save these topologies across laboratories and test stations to provide for error-free setups and drastically reduce the time to set up for a test.

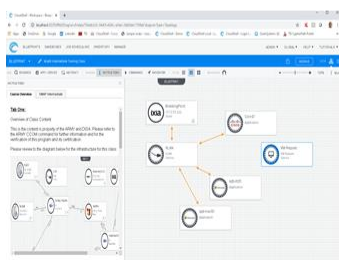
Enterprise Asset Control of Resources

These topologies and assets can be managed and updated globally, as new resources come online or go offline, to afford efficient test resources for the entire organization. Management can view in real time the status of resources and reallocate resources as needed to meet the goals and demands of testing initiatives. Support for multiple tenancies and domains is critical to support the segmented business units and initiatives across organizations as well. Views into the future for planning can be done with automated conflict management for optimal use of lab resources to meet the organization's objectives. Figure 3 highlights the capabilities of the AgileWARE® automated solution that provides for persistent monitoring of the network, alignment of topologies, and automated resource management and scheduling.

Create Test Automation Provision DUTs / Traffic



Topology Setup/CM Resource Allocation



Schedule Executions Coordinate Resources

Name	Status	Actions
Test Case 1	Running	Stop, Pause, Resume
Test Case 2	Completed	View Log, Download Report
Test Case 3	Failed	View Log, Download Report
Test Case 4	Pending	Cancel, Reschedule
Test Case 5	Running	Stop, Pause, Resume
Test Case 6	Completed	View Log, Download Report
Test Case 7	Failed	View Log, Download Report
Test Case 8	Pending	Cancel, Reschedule
Test Case 9	Running	Stop, Pause, Resume
Test Case 10	Completed	View Log, Download Report

Report & Analyze Results, Trends & Metrics



**Figure 3. Capabilities of an Integrated Lab Test Management Framework
AgileWARE® Provides for Enterprise-Level Management of Test and Lab Capabilities**

Higher Levels of Test Automation

With an automated test framework that handles much of the test setup and with an easy-to-use powerful GUI, test engineers can focus on the automation of manual tests, thereby adding complexity and quality to the test process. This automation capability facilitates the reuse of all levels of testing (unit, asset, application, system, regression).

Automated Regression Testing

Libraries of test functions and standardized tests can be built using a test automation framework. These libraries can then be employed to build complex reusable test regression suites to handle formalized test scenarios. These scenarios can then be reused as standardized regressions, greatly decreasing the setup and development time for other test events.

Software-based Scheduling and Coordination of Testing

The scheduling of test resources is highly complicated when multiple tests, multiple test stations, and multiple labs are involved. An automated scheduling tool that factors in priorities and employs rules and preset privileges to guide the scheduling process can greatly increase the efficiency of test labs. Automated scheduling ensures that higher priority tests take precedence, allows tests that run past their allocated time window to be completed, and increases lab throughput.

Automated Test Efficiency Metrics

A test automation framework that possesses a centralized database can capture test performance metrics such as test station or test user efficiency, lab efficiency, cloud usage, project status, test results and many other user-defined parameters. Reports can be easily generated and displayed in dashboard form for real-time analysis of testing performance. These statistics can provide management with powerful insight into the testing process and will allow continuous improvement in the efficiency of its test processes. Figure 4 demonstrates an example of dashboard report capabilities of AgileWARE®.

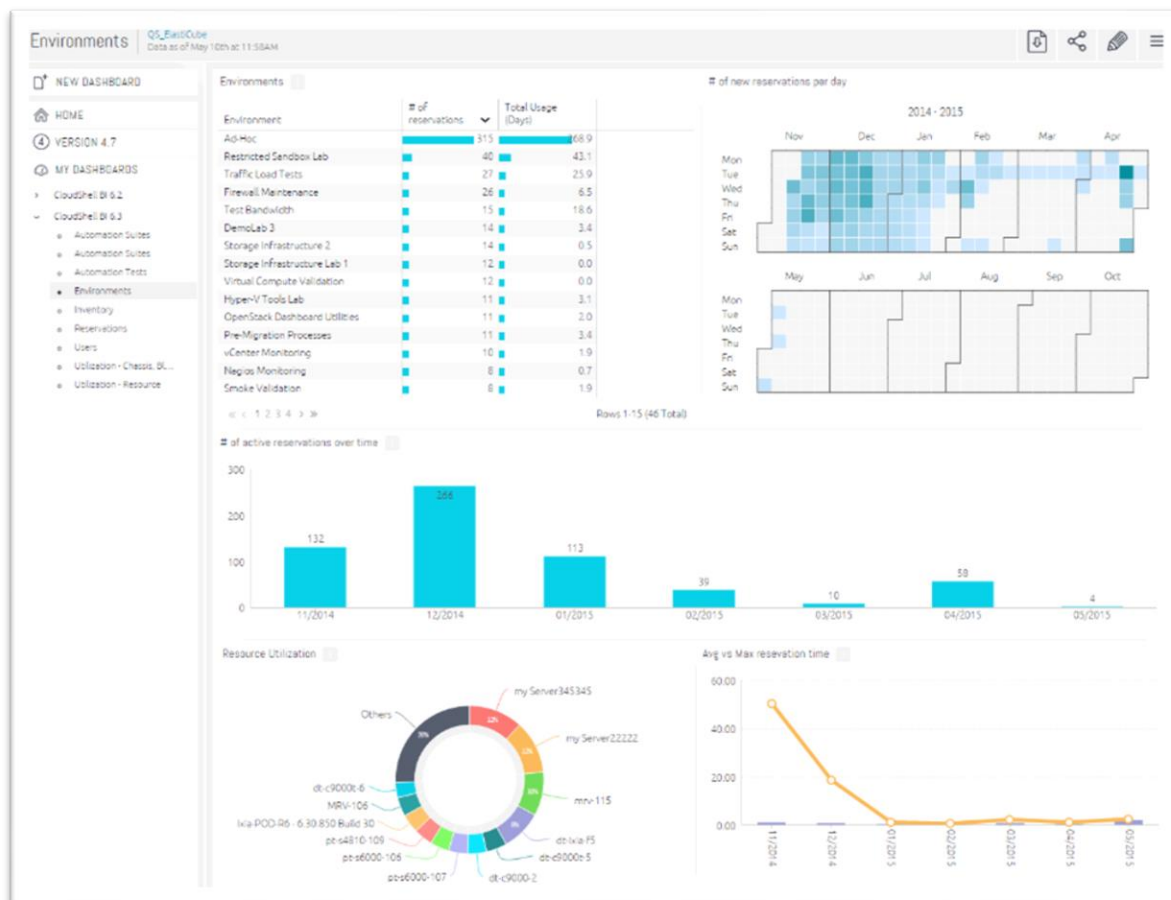


Figure 4. Real-Time Customizable Dashboards and Reports
AgileWARE® Provides Real-Time Reporting of Test Results and Lab Metrics

Enterprise Level Solution

The cornerstone of an automated test system is the ability to support an enterprise level deployment handling the enterprise controls for users, security, IT support and business concerns. This enterprise solution capability provides the security, control, and backup of data and configurations that offer the capabilities previously discussed. Figure 5 graphically portrays this system as it pertains to each view of an enterprise systems which allows the organization to optimize the controls and use from each concern in the organization.

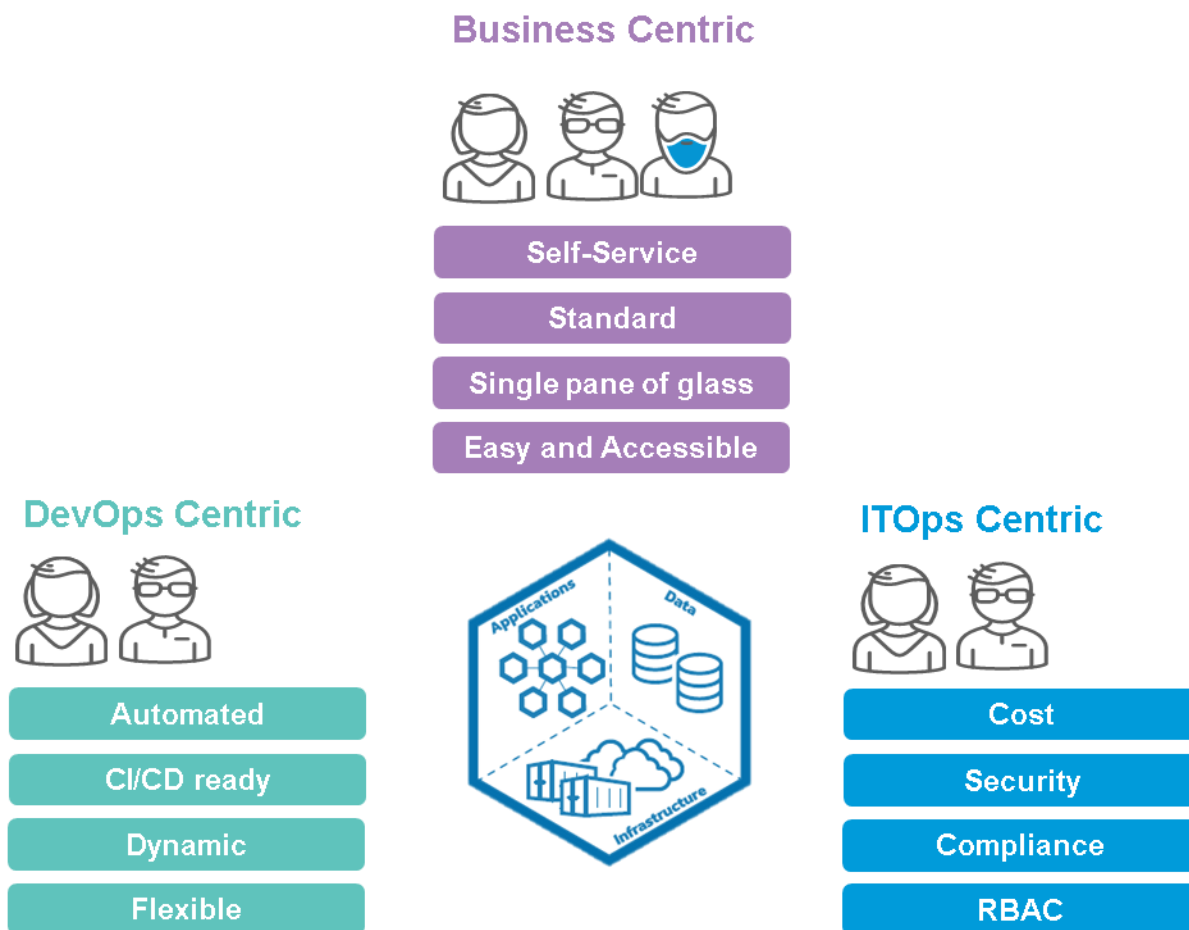


Figure 5. Implementation of Enterprise level Solution
AgileWARE® Provides the Automation and Controls for each aspect of the Organization

Summary

An automated enterprise level test and lab management system will greatly enhance control over test resources in an environment of increasing test complexity. Many well-known industry and government entities such as Cisco, Verizon, Juniper, DISA, Microsoft, Apple, and Tyler Software have adopted CloudShell™ (the tool behind AgileWARE®) to increase their testing and infrastructure consumption efficiencies. Such capabilities provide the ability to better control test configurations, enable enterprise-level control of test/lab assets, allow automated regression testing, provide real-time test/lab metrics, and improve the scheduling, coordination, and throughput of testing/DevOps for faster deployment to production implementations. TSI AgileWARE® has also been deployed in many non-commercial applications (DoD, Federal) for test, infrastructure and workflow management, and cyber exercises.

ANNEX A

Candidate Implementation Plan

Overview

This implementation plan of AgileWARE® is provided for consideration and analysis. The plan provides support for 5 test users managing test resources and 10 other lab users accessing and using the assets within the lab for various tasks (QA, DevOps, Incident Support, etc.). There is 1 lab manager role included.

The candidate implementation is proposed to occur in three phases. The successful conclusion of each phase becomes the criterion for the commencement of the subsequent phase. The successful implementation of a technology shift such as that provided by AgileWARE® requires a commitment to a culture change; therefore, education sessions, formal training, and advisory services are included in this proposed plan.

Site Survey

An on-site equipment and facility audit is proposed to verify and validate the lab environment at the proposed location for implementation. The audit will determine the number of test development engineers, test stations, test equipment types and quantities, and specific needs of each individual lab. These results will determine overall system hardware, software and training requirements.

Phase 1 – Installation of Software and Initial Training

- Install a CloudShell™ Enterprise Server with 15 reservation clients, 2 Authoring clients and 2 Runner clients, Save & Restore and Business Intelligence server tools.
- Validate installation and setup of clients within each lab; verify client access to server.
- Deliver on-site training for lead test development and resource management personnel. Train test engineers on the setup of a function library of test assets and scenarios. Develop standard report formats for automated report generation.
- Define attributes to drive performance metrics.
- Identify the requirements for drivers of nonstandard or unsupported test equipment resources to add them to the test topology.
- Identify tests and the resources required to start defining resource and function libraries.

Phase 2 – Development of a Library of Tests and Regressions for Use in Typical Testing Scenarios

- Further refine and develop additional needs-based test cases that match repetitive functions that are modular in nature and may be used as building blocks of varied and more complicated test suites.
- Design the employment of the Business Intelligence server to provide metrics on test and infrastructure performance important to the customer.

- Implement dashboards to visualize test performance metrics and generate test performance reports.
- Develop and verify the use model with some typical tests from one or more target labs.
- Identify additional training and consulting needs for the next phase: shells that must be developed; resources that must be defined; and function libraries that must be built.

Phase 3 – Additional Training

- Provide additional training and advisory services as required to ensure successful implementation and full utilization of CloudShell™ capabilities.

Pricing for Candidate AgileWARE® Implementation

The prices below are provided in support of the candidate installation for year 1 for the implementation. Training and consulting charges are expected to be less for additional installations.

No	Product	License Type	Unit Price	Qty	Extended Price
1	Enterprise Edition Server <ul style="list-style-type: none"> • Supports unlimited clients • Includes HA and full job Scheduler and execution module 	Subscription	\$54,000	1	\$54,000
2	Authoring Client	Subscription	\$8,640	2	\$17,280
3	Execution Client	Subscription	\$5,184	2	\$10,368
4	Active Reservation Client	Subscription	\$3,000	15	\$45,000
5	Save & Restore option	Subscription	\$36,000	1	\$36,000
6	Business Intelligence Server	Subscription	\$36,000	1	\$36,000
8	Total software licenses:				\$198,648
9	25 days of professional services for installation, coaching, initial function library startup and basic configuration	N/A	\$2,100 per day	25	\$52,500
10	Admin training (2 days)	N/A	\$3,000	1	\$6,000
11	Authoring training (5 days)	N/A	\$17,000	1	\$17,000
12	Total for product and services				\$274,148

No	Product	License Type	Unit Price	Qty	Extended Price
Notes: <ol style="list-style-type: none"> Enterprise Edition Server does not include a database license (MS SQL required), or hardware for server/client use, or base operating software for server or clients. Annual Support Program is included in subscription and includes: <ul style="list-style-type: none"> Free software upgrades Online support: Web portal and Help Desk Support Phone/e-mail support during regular work hours (9 am to 5 pm) Monday to Friday Integration with Test and Measurement equipment and new shell development (dependent on the results of the site survey) are not priced in this quote. 					

ANNEX B

Return on Investment Results for Candidate Implementation Plan

Overview

This annex provides the Return on Investment (ROI) analysis for the candidate implementation of AgileWARE®. The analysis assumes that 5 test engineers and 10 lab users are supported at the total weighted costs per each outlined below:

Number of lab users	10	\$150,000
Number of lab managers	1	\$200,000
Category 1: Test interns	1	\$60,000
Category 2: Test technicians	1	\$100,000
Category 3: Test engineers	2	\$140,000
Category 4: Automation engineers	1	\$180,000

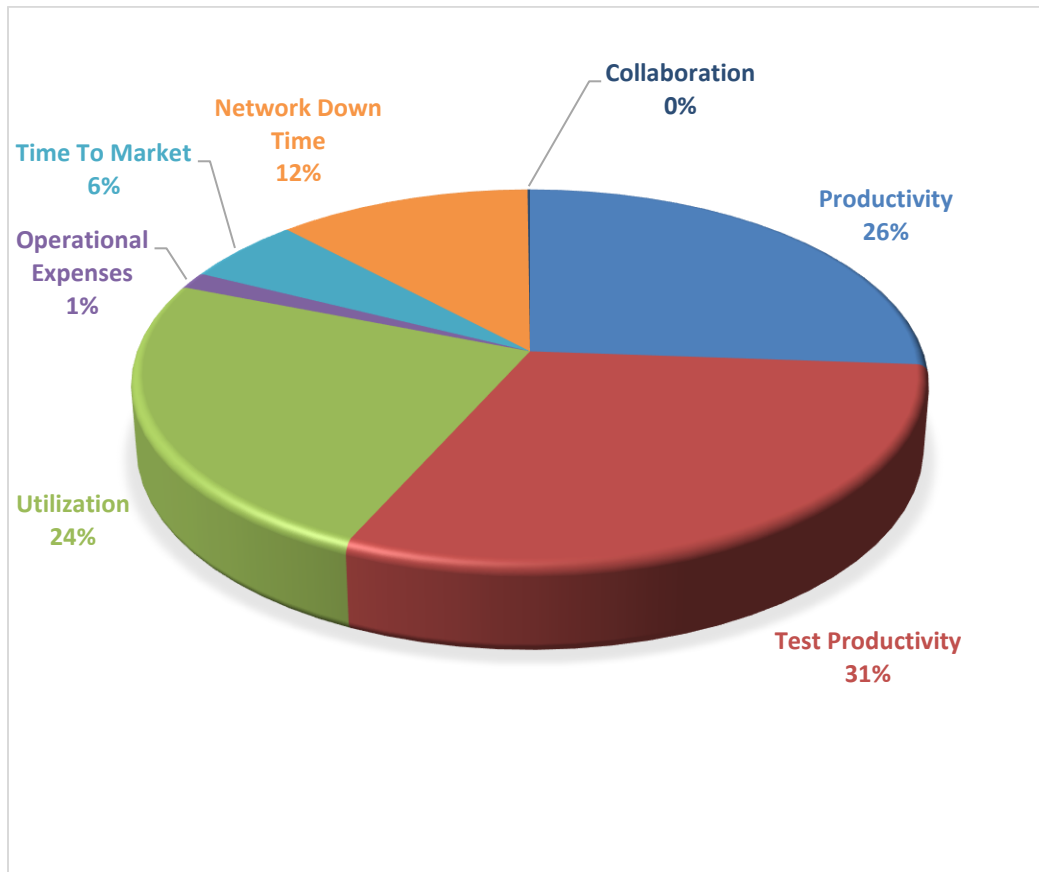
Testing efficiencies, which were determined through the historical analysis of industry implementations of AgileWARE® are captured as dollar cost savings. These efficiencies can be used to increase test throughput or to reduce the manpower required to run test operations. The table below provides a summary of the benefits of the implementation of the candidate system.

Financial Benefits

The table below shows the benefits provided by AgileWARE® per each year. These benefits are grouped into seven categories.

Benefits	Year 1	Year 2	Year 3	Total
Productivity	\$330,000	\$330,000	\$330,000	\$990,000
Test Productivity	\$310,577	\$425,000	\$425,000	\$1,160,577
Utilization	\$172,500	\$301,875	\$443,756	\$918,131
Operational Expenses	\$18,000	\$18,000	\$18,000	\$54,000
Time To Market	\$70,725	\$70,725	\$70,725	\$212,174
Network Down Time	\$150,388	\$150,388	\$150,388	\$451,165
Collaboration	\$1,442	\$1,442	\$1,442	\$4,327
Total Benefits	\$1,053,632	\$1,297,430	\$1,439,311	\$3,790,374

Cumulative Percentage Breakdown of Benefits over 3 years



Better Productivity Financial Benefit

With its point-and-click applicative environment, AgileWARE® enables improved efficiency for time-consuming tasks performed by human resources. Test implementation, integration, and debugging are done faster, while overhead activities such as reporting, data collection, and documentation are automatic. The system's topology, based on a central repository of tests and test results, enables reusability, thus saving effort of recurring time-consuming tasks. The numbers in the tables below represent actual use data gathered from typical customers using the solution above and reflect typical improvements. Actual improvements may vary based on each customer's environment, product(s) and other considerations.

Productivity Improvement Areas

Lab user productivity (average hours for one user in one week)	Today	With CloudShell	Productivity Increase
Topology creation time (hr)	1	0.3	70%
Time to provision all devices in a topology (hr)	2	0.05	98%
Time searching for available resources (hr)	1.25	0.05	96%
Time to resolve conflicts (hr)	1	0.01	99%
Time to debug and fix incorrect setup (hr)	0.5	0.5	0%
Time to reproduce a previously created topology with provisioning (hr)	2	0.1	95%
Time to produce lab reports (hr)	1	0.1	90%
Total hours per week	8.75	1.11	87%
Total hours per year	455.00	57.72	87%

Test Productivity Improvement Areas

User productivity (average hours for one user in one week)	Today (hours)	With TestShell (hours)	Reduced Hours	Productivity Increase
Development of new tests (User Categories 1-4)	6	4	2.00	33%
Test setup and configuration time (User Categories 1-4)	6	0.2	5.80	97%
Maintenance or adoption of existing tests (User Categories 1-4)	4	2	2.00	50%
Data collection (User Categories 1-4)	8	1	7.00	88%
Report generation (User Categories 1-4)	8	0.2	7.80	98%
Data analysis, trend reporting (User Categories 3-5)	4	0.2	3.80	95%
Totals per user	36	7.6	28.4	79%

Better Quality Financial Benefit

The consequences of poor quality are expenses dealing with replacing or fixing units, shipment of units, and indirect lost income caused by unsatisfied customers. Improving quality could be achieved mainly by:

- Better test coverage: Using AgileWARE® to enable the acceleration of test implementation, thereby enabling better test coverage.
- Availability of real-time information: The automatic aggregation of test results into a central repository, combined with Business Intelligence abilities, enables better decision-making based on real-time information derived from the tests.

Reduced Cost of Poor Quality		1-Year Save	3-Year Save
Estimating the cost of poor quality			
Estimated cost of total test development per year	\$620,000	N/A	N/A
Estimated percentage of cost of poor quality	3%	N/A	N/A
Cost of poor quality per year	\$18,600	N/A	N/a
Potential Reduction in Cost of Poor Quality		1-Year Save	3-Year Save

Attributed to increased test coverage	15%	N/A	N/A
Attributed to availability of real-time information	3%	N/A	N/A
Total		N/A	N/A

Note that these numbers above are not included in the financial ROI calculations but shown only here for reference.

Better Automation Financial Benefit

Testing is an essential part of a product's life cycle. The testing phases take time and resources and might become bottlenecks that cause test project overruns and increased time to deliver the results. The testing phases could be optimized mainly by:

- Better test-resource allocation: A centralized resource management seamlessly integrating with the test stations enables optimizing the use of resources.
- Better automation of test execution: With AgileWARE® test execution of multiple stations could be scheduled and managed, thus maximizing tests and resources utilization.

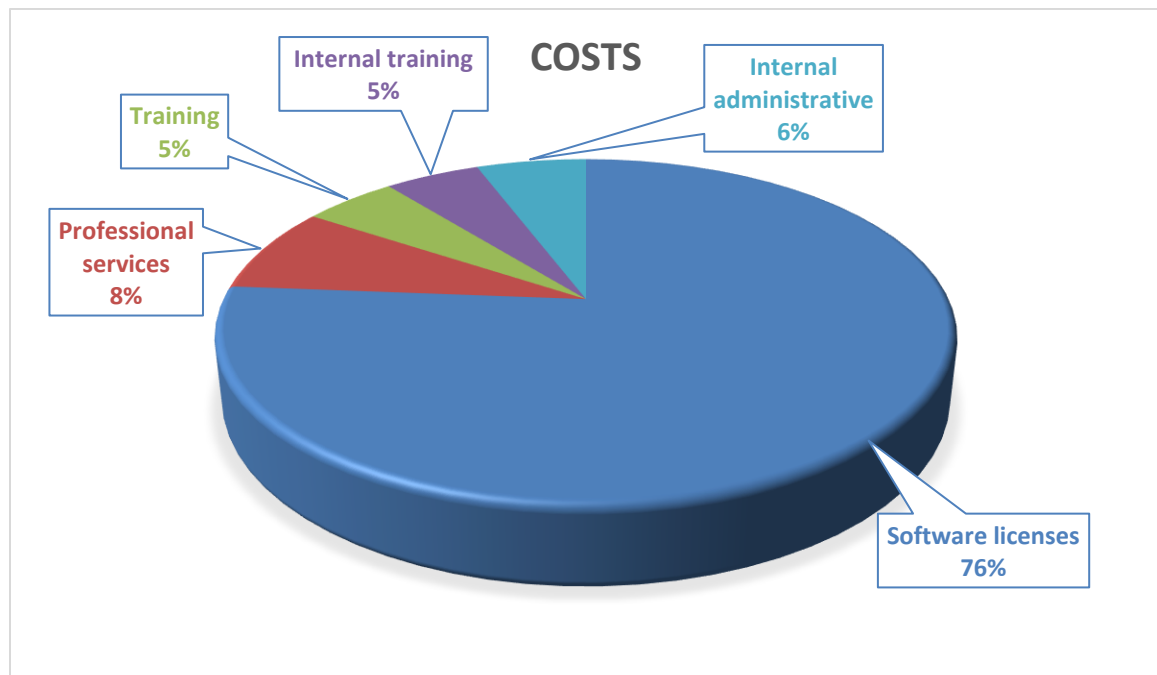
Reduced Cost of Project Overruns		1-Year Save	3-Year Save
Estimating the cost of testing time overruns			
Average cost of delay in testing results delivery per month	\$333,333	N/A	N/A
Average delay in testing results in months	0.5	N/A	N/A
Total cost of delay in testing results delivery	\$166,667	N/A	N/A
Potential Savings		1-Year Save	3-Year Save
Attributed to optimization of test resource allocation	25%	N/A	N/A
Attributed to automation of test execution management	25%	N/A	N/A
Total		N/A	N/A

Note that these numbers above are not included in the financial ROI calculations but shown only here for reference.

Cost

The table and charts below show the projected costs associated with the deployment and assimilation of AgileWARE® over a 3-year period. Costs comprise both pre-start and recurring costs during the first 3 years of use. The costs include all general costs associated with deploying the solution (software license, support, training, administration, etc.).

General Costs	Year 1	Year 2	Year 3	Totals
Software license cost	\$198,648	\$198,648	\$198,648	\$595,944
Professional services	\$52,500	\$0	\$0	\$52,500
Training	\$23,000	\$0	\$0	\$32,380
Total general costs	\$274,148	\$198,648	\$198,648	\$671,444
Internal Costs	Year 1	Year 2	Year 3	Totals
Internal training cost	\$32,693	\$0	\$0	\$32,693
Internal administrative cost	\$17,692	\$10,000	\$10,000	\$37,692
Total internal costs	\$50,384	\$10,000	\$10,000	\$70,385
Total costs	\$324,533	\$208,648	\$208,648	\$741,829



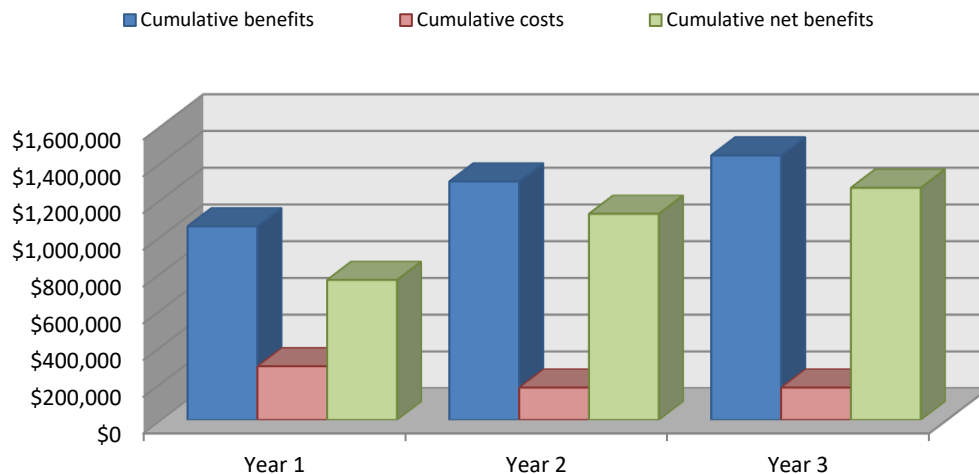
The table below shows the total net benefits and Return on investment.

Financial Analysis	
Annual return on investment for first year	325%
Payback period (years) - includes startup time considerations	0.3
Payback period (weeks)	13.0
Total 3-year net benefits	\$3,048,545
Average annual cost of ownership	\$247,276
3 Year Cumulative ROI	511%

The table and chart below represent the total 3-year net cumulative benefits. The total 3-year net cumulative benefits equal the cumulative benefits minus the cumulative costs.

	Year 1	Year 2	Year 3
Cumulative benefits	\$1,053,632	\$1,297,430	\$1,439,311
Cumulative costs	\$324,533	\$208,648	\$208,648
Cumulative net benefits	\$729,099	\$1,088,782	\$1,230,763

Annual Net benefits



Cumulative Financial Table Results

Annual return on investment (1 st year)	325%
Payback period (weeks)	17
Cumulative return on investment	511%
Benefits (3 Years)	
Increased Lab Productivity	\$990,000
Increased Test Productivity	\$1,160,577
Increased Utilization	\$918,131
Decreased Operational Expenses	\$54,000
Reduced Time to Market Expenses	\$212,174
Reduced Network Down Time	\$451,165
Reduced cost of collaboration	\$4,327
Total benefits	\$3,790,374
Costs (3 Years)	
Software license cost	\$595,944
Professional services	\$52,500
Training	\$23,000
Deployment and Administration Costs	\$70,385
Total costs	\$741,829
Total 3-year Net Benefits	\$3,048,545