

Maximizing the Value of Your Test Environment



Dan Guss - Ameritech

I. OVERVIEW

This document accompanies the presentation Maximizing the Value of Your Test Environment presented by Dan Guss of Ameritech at the STAR 99 conference.

The overall objective of this presentation is to provide insight to the integrated processes, financial and resource requirements, and technical components necessary to achieve a robust end-to-end test environment. A fully integrated environment will enable end-to-end testing, quality improvement and pre-production certification for your suite of applications. The presentation further examines the vision, components, approach and methodology, and business processes that are critical to a successful test environment. This paper contains the following sections:

- Background Provides context regarding Dan Guss' background and his latest challenge to build a robust end-to-end test environment for Ameritech
- Understanding the Business Opportunity Defines the attributes of the business opportunity and reviews some the challenges (chaos) often encountered when a mission critical opportunity is presented
- Setting the Business Vision Reviews four key areas which must be considered in order to achieve the program objective
- Creating the Total End-to-End Business Solution Examines the business processes, financial and resource requirements, and technical components in more detail necessary to create a total end-to-end solution and presents several insights to insure you maximize the value of your test environment
- Conclusion Revisits the four key areas which must be considered in order to achieve the program objective

II. BACKGROUND

Dan Guss has over 16 years of large system development and testing experience at Ameritech, a major telecommunications provider in the Midwest. He has provided leadership to many projects in roles including program manager and chief technical architect. He has extensive experience in the areas of capacity planning, systems monitoring, and network architecture. Dan recently concluded a project where he was responsible for building and managing a complex end-to-end computing environment to test over 40 of Ameritech's most mission critical applications which span the ordering, provisioning, billing and maintenance business functions.

III. UNDERSTANDING THE BUSINESS OPPORTUNITY (SLIDE 3)

Today there are many complexities in corporate information technology (IT): centralization, decentralization, insourcing, outsourcing, and rightsizing just to name a few. Through all of this change, quality of service has been inconsistent as new business units, functional areas and business partners have emerged. Consequently your team may not have complete control of all aspects of their responsibilities. Your data centers may be outsourced, you may have applications that are vendor built and supported, your technical architecture may incorporate middleware that is critical to the operation of the application. The information systems of today are more decentralized and incorporate many more components than the systems of just a decade ago.

Imagine the challenge of building an end-to-end testing environment encompassing all the components required to execute complete business functionality. How would you approach this opportunity?

This presentation and paper provides insight into these questions to help you maximize the value of your test environment. A business opportunity to build an integrated, end-to-end test environment is an exciting and challenging journey to undertake. This type of opportunity was first presented during the initial start-up phases of a program at Ameritech when there was a potential for 50 to 200 applications hosted by 10 to 15 different platforms. In addition to the technical complexity and magnitude, the following attributes accompanied the business opportunity in October of 1998:

- The financial resources were limited to a fixed amount of budget allocated to the project.
- The timeframe was very aggressive with four months given to the build of the environment and eight months for the execution of the test. At times, the build and the execution overlapped for specific applications. This presentation focuses primarily on the build efforts. While four months does seem to be a reasonable amount of time to construct an end to end environment, this timeframe really is very aggressive when you consider tasks such as requirements definition, architecting the environment, request for bids, vendor contract negotiations, hardware and software procurement, hardware and software installation, application installation, and environment validation.
- The resource availability was constrained due to the lack of resources with skills sets required to build and support the environment.
- Support was required by a number of independent groups. The test relied on a number of different individuals from eight to ten in-house organizations, three to five major application vendors and countless 3rd party software vendors to create and implement an end-to-end technical solution.

Maximizing the Value of Your Test Environment



Dan Guss - Ameritech

Someday an opportunity like this may be presented to you or your group. The complexity or magnitude may differ, however, all the insights discussed in this presentation can be applied to many different situations. For example, you may be responsible to build a new standalone application test environment, stabilize or improve an existing test environment or to consolidate several independent test environments into a single, integrated test environment.

With all the unknowns and challenges surrounding an opportunity, it is important to quickly create structure and set a vision. Strict financial constraints, unclear technical requirements, intricate political landscape, fluctuating scope, undefined deliverables, reliance on vendor relationships, team dependencies and unclear resource requirements are some of the challenges which may accompany this opportunity.

IV. SETTING THE BUSINESS VISION (SLIDE 4)

The business vision is not something to hide. The vision for the technical, or Environment Team, should be well documented, merged, and presented with the visions from peer groups on the goals of the program. At a minimum, this vision should be shared with the key organizations that will be involved with procuring, building, supporting and using this environment. Identification of the objective, components, resources, and deliverables will position the program for success. Kick-off meetings will jumpstart the project, initiate the face-to-face bonding, and promote buy-in to your vision. It is important to listen and act upon the input received from these initial gatherings to insure the partnership is reinforced from day 1.

When presented with such an opportunity to build an end-to-end test environment, the tendency may be to focus in on the required technical components and ignore several other critical areas. Many programs have gotten off on the wrong foot only later to find critical errors in their initial planning.

Setting the business vision to create a robust end-to-end test environment involves four main areas: Integrated processes, financial requirements, resource requirements, and technical components. The following section entitled 'V. Creating the Total End-to-End Business Solution', explains each of these areas in more detail.

V. CREATING THE TOTAL END-TO-END BUSINESS SOLUTION (SLIDES 5 - 15)

Following is a breakdown of each area required for creating the total end-to-end business solution.

Integrated Processes

In order for the program to be successful, many processes must be defined and integrated across teams in order for task execution to be as effective and efficient as possible. Two activities that encompass integrated processes that are important to understand are scope definition and program management.

Scope definition is critical to the environment build process. Wide fluctuations in scope can hamper the team from developing the focus they need to make progress in achieving the objective. For example, the initial scope of the Ameritech program included 176 applications. Therefore, the requirements team was asked to begin investigation of 176 applications. Fortunately the scope was quickly whittled down to approximately 70 applications, which was much more realistic. It was clear that during the initial stages of the requirements process when 176 applications were being considered, time was being wasted, political barriers were being formed and the credibility of the project team was being impacted. It was evident that at least a portion of the work being conducted would be throwaway. The initial scope reduction increased the focus of the team tremendously.

The Environment Team should participate in scope definition sessions in order to influence the direction by disclosing technical challenges and implications. Other teams may not be aware of the lead-time and cost involved in purchasing additional hardware. Timeframes and cost may alter the reality of the situation if the requirements are not clearly defined in the beginning.

For example, during the Ameritech program, the test team had difficulty identifying the data test beds that would be used during executions. The test data selection needed to occur prior to finalizing the disk requirements because the two test beds under review contained extremely different volumes of data. The test team did not understand this dependency nor did they understand the impact of a delay in their selection. Once the Environment team explained the potential difference in cost and the lead time of nearly two months required for ordering the hardware, which included design, quote, contract negotiation, procurement and installation activities, the test team realized the true urgency in their selection activity.

A second activity regarding integrated processes is program management. The larger the scope of a program, the greater of importance is program management. An initial, high priority activity should be to create a high-level project plan. The project plan should be focused around the deliverables defined in the deliverable map. Deliverables provide a paper trail of what has been done to date, including clear documentation of any assumptions or constraints in the environment build. By determining the required

Maximizing the Value of Your Test Environment



Dan Guss - Ameritech

deliverables at the beginning of the program, they also help to establish milestones, and help drive your ability to provide the solution. The list of required deliverables may be modified as progress is made and the end objective becomes clearer.

As shown on slide #6, a deliverable map can be useful to identify key deliverables for each phase. This encourages one to begin with the end in mind and motivates forward thinking. It also helps to define the overall project objective and direction. The map becomes a communication tool when combined with the maps from any peer teams such as Test Management or Program Management. A single map provides an overall view of the program on one page.

Deliverables on the Deliverable Map are used as input into the Integrated Timeline that will allow dependencies across teams to be documented. Environment tasks must be added to the test plan, which is the first step to creating an integrated timeline. Another helpful task is to perform scenario planning to detail the test execution activities. An execution model is useful for everyone to have visibility to the overall schedule and the activities that need to occur. Execution results should be tracked and fed back into the model to increase the accuracy of the estimates.

Another important aspect of program management is quality assurance (Q. A.). Q. A. reviews are beneficial to get an objective viewpoint of the work that is being conducted. These should be scheduled on a regular basis and attended by the leadership from each team.

Financial Management

While it is obvious that a sufficient budget is required to build a testing environment to meet the requirements, many projects have fallen on hard times because the cost of several less-than-obvious components were not taken into consideration. Team architects must be challenged to expose all required components as early in the program as possible. These components must be reviewed in detail to identify all potential costs relating to them. Conservative estimates should be created because, undoubtedly, there will be hidden costs that are not incorporated in the initial design. The lead times to allocate funding, negotiate contracts, and procure the components need to be well understood if the program is going to remain within the aggressive schedule.

The Environment Team should challenge all the requirements presented with the program scope. There may be some opportunity to reduce the scope minimally and save a substantial amount of money. For example, during the Ameritech program, two platforms were eliminated that had minor test cases and the program budget was reduced by several million dollars. The test cases for these platforms were validated using other acceptable methods.

Resource Requirements

The three key areas of focus regarding resource requirements are organization, workplans and partnerships. It is important not only to determine which resources are at your disposal, both inside and outside of your immediate organization, but also to define clear responsibilities such that completeness is obtained and overlap is reduced.

When designing the organization, it is important to insure that the structure is flexible. As the program matures, the team's focus, and therefore its organization, should adapt to the critical tasks of each phase. Minor reorganizations can greatly improve the effectiveness and efficiency of the team members. A team composed of individuals with broad, diverse technical competency will be able to drive the best solution, examine the costs and associated risks, and offer options for consideration.

The next slide shows an example of an organization chart with clearly defined roles for each team. The Core Team, which leads the test effort, could be arranged into 3 sub-teams: Program Management Office, Test Management and Environment. The Program Management Office team is responsible for managing program budget, staffing, status and quality. The Test Management team is responsible for developing the test strategy and performing test execution. The Environment team is responsible for planning and ordering the environment, building the environment and supporting the environment. Although each team has their own defined set of tasks, they do not operate independently. In order for everyone to be successful, they must integrate their tasks and become interdependent and foster the one team concept. This partnership will allow you to reduce conflict and work together to achieve results. The integrated timeline is a tool that will require all teams to gain insight to their individual issues and drive out potential conflict before it arises.

A method to manage resource assignments is the creation of work plans. Work plans should be developed within each team to document low-level tasks. Critical activities that require near daily attention by the management team could be tracked in duplicate on a "Hotlist" noting task description, assigned resources, and due dates. Hotlist reviews should occur two to three times per week by the leadership team.

Communication and partnership with the other participants of the testing organization are critical for the Environment Team to be successful. A complete understanding of the entire business solution, including resource planning, financial planning and integrated processes are key to the success of the program. Clear and open communication lines between groups should be established from the beginning. Expectations and service level agreements should be documented and agreed upon by all parties. Also, it is important to engage and leverage vendors with the right core competencies.

For example, Ameritech has outsourced data center operations. The outsourcing contract with the vendor to build new environments estimates 29 weeks from initial design through validation and environment turn over. However, the program timeline allowed only 16 weeks for the environment build activities to occur. By partnering closely with this vendor, the environment

was actually constructed within the extremely aggressive schedule. The vendor supplied several resources to work with the core team onsite to help manage the integrated activities at a very low level of detail. This close partnership identified some shortcuts to be taken that either eliminated or significantly reduced several critical path activities. The end result not only met the program timeline but also provided insight to streamlining the overall environment build process, which will save Ameritech and the vendor time and money going forward.

Technical Components

The test execution environment will be the most visible deliverable by the Environment Team. Therefore, you must take care to insure a complete environment and infrastructure is created, otherwise the test will not occur. The tendency may be to focus in on the application hosts required for test execution with minimal attention paid to the other components. In addition to the base configuration required for test execution, additional components could prove invaluable to providing a stable environment or enhanced testing capabilities.

For example, although it may not be a requirement from the Test Management team, the Environment Team may want to implement production-like monitoring capabilities so that the environment support activities are proactive instead of reactive, waiting for the test team to call when a failure occurs. Not only will this capability promote reductions in the failure rate, it will provide insight into changing capacity requirements of the applications so that production capacity can be adjusted accordingly, for example, prior to a release being migrated into production. In order to create this level of monitoring, it is critical that the requirements team work with the application experts to identify critical system and application processes to be monitored as the base requirements are gathered.

Initial activities to develop the technical environment for an end-to-end business solution include planning and ordering the required components. Once the components are delivered, the environment must be built to insure all test requirements are met. Once all components are installed, a validation of the technical components should be performed prior to the test execution. An implementation life cycle should be followed such as Plan, Order, Build, and Support.

The following is an overview of several technical components that should be considered when building an end-to-end testing environment:

- Project Team Infrastructure

The testing organization needs to have a reliable desktop and server environment that provides e-mail, data sharing, directory management, data backup, and security. Process automation capabilities such as release management, defect tracking and software distribution would aid in improving the effectiveness of the project team as well.

Maximizing the Value of Your Test Environment



Dan Guss - Ameritech

- Application Environment

The application environment refers to the test execution environment. The end-to-end testing environment built for the Ameritech program was constructed to be as much like the production environment as possible. The environment complexity is directly related to the unique number of hardware and software components that must be integrated to support complete business functionality.

- Monitoring and Environment Management

The monitoring and environment management component is made up of development and production servers supporting your test management capabilities. These tools can help in proactive error detection, problem correlation, trend analysis, capacity planning and overall performance monitoring and tuning.

- Application and Data Staging

Applications are in constant state of change. Code fixes, releases, and 3rd party software upgrades are just a few of the items that are often in flux. The servers that support this function would be used to manage releases, validate changes, stage production data, house data tools, and provide other control functions.

- Test Preparation Lab

Test preparation capabilities could include tools to plan test scenarios, create test data, and generate automated test scripts. When considering creation of automated test scripts, the components utilized during script generation must mirror the actual test components to avoid potential inconsistencies which could render the test scripts useless.

- Test Scripting and Automation Lab

The scripting tools have come a long way and are providing major benefits to organizations that have leveraged their capabilities. Tools can be integrated into the testing process and require a high performance environment. These servers are used to simulate volume tests and highly repeatable processes.

- LAN/WAN

The technical components will need to be connected with network technology. Multiple protocols may be required for more complex environments. Firewalls could be extremely useful in isolating the test environment from production.

VI. CONCLUSION (SLIDE 16)

The overall objective of this presentation was to provide insight to the integrated processes, financial and resource requirements, and technical components necessary to achieve a robust end-to-end test environment. By creating a fully integrated environment, your test organization will have a fully functional environment that will allow for end-to-end testing, quality improvement and pre-production certification for your suite of applications.

To recap, the following points are critical to building an end-to-end business solution to maximize the value of your test environment:

- Manage project by defined scope, deliverables and scheduled milestones
- Be involved in the financial management and tracking of the technical environment throughout the project
- Build a flexible team with strong technical competencies and processes, and empower them
- Implement a complete, robust computing environment including all critical components