

CVS.com

Building an eBusiness Test Group

Jim Hancock

Quality Assurance Director

Presented at the SASQAG on June 15, 2000

Welcome to everyone.

Introduce yourself...

Overview

- **The Basics**
 - Staffing Considerations
 - Test Environments
 - Client Side Testing
 - Tools
- **Processes**
 - Project Migration
 - Bugs Tracking
 - Testplan Creation
- **Additional Considerations**
- **Must Do List**

I'm planning to discuss the following areas:

What I call the basics ...this includes.....

Staffing

- Skill Sets
- Training
- Team Structures

With regard to staffing we will discuss staffing issues including:

1. Some of the necessary skill sets to build a solid team.
2. What training is important.
3. How to build a solid team structure.

Staffing - Skill Sets

- Specific Skills
 - Lead Experience
 - Database Knowledge
 - Coding Skills
 - Internet Skills
 - QA Skills
 - Motivation / Good Attitude ...goes a long way!
- Newer versus Older Company
- Putting the puzzle together

Choosing a good staff is the most important task you will ever have. Your staff will make you or break you.

The basic things to look for are a good attitude and a desire to learn and grow. Beyond that you should fill out skill sets in the following ratio (these can overlap):

x 1/3 to 1/2 Lead

x 1/3 to 1/2 Database Testers

x 1/6 to 1/3 Coding Skills (for automation testing)

x Good QA skills are essential for everyone.

x Internet skills are good (less ramp up time), but a good tester can pick up a new environment fairly quickly. You should have at least one person who knows Browsers well though...

You will want more technical skills in a start-up staff than will be necessary later. This is because you are bootstrapping the team and need to build from scratch.

Distribute the skills as makes sense. Some test areas need more database ...others need more leads because there may be many smaller projects... and so on.

Staffing - Training

- Leads - Mentoring
- Database - Training (BCC/UW)
- Coding Skills - JavaScript, VB, Scripting Tools
- A Good Training Program is Essential!
 - Provides growth for staff
 - Makes your team stronger
 - Makes you more competitive as an employer
- Try to promote from within

Leads - Should start out with strong communication and planning skills. Mentoring is a really good way to allow growth in this area. The main thing to teach is how not to want to make everybody happy.

Strong Database skills are essential to the success of the team. Always approve database training! I recommend BCC or the UW, but there are many sources.

Coding skills are also important. The better a tester understands what it is they are testing, the better they can break it.

Also, don't send someone with poor coding skills to automation training ...the ramp up to get to good quality scripting is a long time. Send them to VB or ASP first and see if they like to code ...then spend the money on tool training ...it is a win/win because even if they decide they don't like to code they will understand what they are testing better.

Train, train, train ...I think this is self evident, but if you are not learning something new, you are falling behind.

Try to promote from within ...this may lower your attrition and make your staff happier.

Staffing - Team Structures

- Considerations
- Skill Sets
- Personalities
- Lead Ability

CVS.comQA is divided into the following testing areas:

- Transaction / Data Exchange / Membership
- Product Catalog
- UI/Content
- Test Automation
- Plus additional projects are assigned in line with this...

Basic things to keep in mind:

- Match skill sets to needs in each area.
- Try to match personalities within each group.
- Ensure sufficient leadership ability in each group.
- It is better to overstaff leads than under-staff, since it allows leads to do some level of testing and keep their hand in things (and in case of attrition)
- Have leads in charge of other things as well, such as the test lab, test-planning, bug tracking after they are set up ...otherwise you will over extend yourself.

Test Environments

- Technical Structure (Microsoft)
- QA Perspective
- Components
- Other Considerations
- Client Side Testing

The technical (server) structure follows basic Microsoft technology and we will touch briefly on the basic components.

There are any number of QA server environments that can be set up and we will discuss these at some degree.

There are many different kinds of components that migrate through the various QA environments and we will discuss the primary ones (for an MS shop).

There are a number of other things to consider, the most important being scheduling the moves through the various environments.

Finally, good client side testing is absolutely essential for internet testing, since the client could be using any number of different browsers, operating systems, network connectivity, etc.

Test Environments - Technical Structure

- BigIP
- IIS Servers
- SQL Server
- LDAP and Membership
- MSMQ

CVS.com uses Microsoft technology including the following main components:

- BigIP - Distributes customer load among various IIS servers.
- IIS Servers - Serve up ASP generated HTML pages, graphics, etc.
- SQL Server - We use SQL server database to track transactions and maintain our product catalog (up to 8000 products now last count).
- LDAP and Membership - Maintains all of our customer data.
- MSMQ - Serves as a message queue to hold and send in batch mode accumulated transactions to the fulfillment center in Ohio.

Test Environments - Components

- ASP/HTML Pages
- Images
- Stored Procedures
- Database Structure (tables, columns, etc.)
- Installed Components

The following types of things move through the various QA environments before making their way to the production release:

ASP/HTML Pages - ASP is typically VB Script embedded in the HTML page and interpreted by the IIS server. It generates dynamic HTML pages which are sent to the browser.

Images - Many many images will be added and altered. These all need to stay in synch with the pages they are contained in.

Stored Procedures - Much of the processing is done within the database in the form of stored procedures. This is usually to minimize the network traffic and speed up transactions. These will change and be shared by multiple projects and need to be fully tested out whenever changed.

Database Structure - Tables and columns will change and be added. This is hard to keep in sync with page changes many times and can create code management issues without a good code management tool.

Installed Components - Many times components will be installed directly on a server for things such as search engines. These take the form of exe's and dll's. Always load test these ...one more time ...always load test these!

Test Environments - QA Perspective

- Development Environment
 - Project level granularity (+ prototyping)
 - Very very dynamic
- QA Environment
 - Project level granularity
 - Change twice a day
- Staging Environment
 - Release level granularity
 - Change twice a week
- Production Staging
 - Installation test /release dry-run
 - Should be fully scripted
 - Change once a week
- Production
 - Should be fully scripted
 - Change once a week

The following are the number of environments recommended (based on a weekly release cycle):

- Development - QA never looks at this environment. It is constantly changing.
- QA Environment - QA gets a first look at a project in this environment.
- Staging Environment - This is primarily for system level testing and User Acceptant Testing (UAT). This is the “gold” release to production including only those projects meeting release criteria at the agreed upon deadline.
- Production Staging - This allows the build team and DBAs an environment to do a dry run prior to the actual production release. QA should BVT after the installation is complete.
- Production - This is live! Releases should be limited to once a week if at all possible, since otherwise the overhead of the actual release process becomes overwhelming. Finally, always test the release ...on every server...with various browsers ...and using AOL and Mac.

Test Environments - Other Considerations

- Release Schedules
 - “Code Complete”
 - “Code Freeze”
 - “Go/No Go” Meeting

Potential Scheduling Issues

- Being Firm (QA MUST have final decision to release)
- Hard Release Dates (finish early)
- Merging Large Projects

Release schedule should be set for projects long before they reach QA and of course QA should be involved from the start.

Some terms are not entirely understood, so I'll enunciate:

Code Complete - All major code is developed and unit tested. The only code changes after this date should be bug fixes.

Code Freeze - It is time for a final regression pass ...if there are bugs found after this point they need to be reviewed individually and unless they are showstoppers will remain open until after release.

“Go/No Go” Meeting - Determine which projects meet release criteria and which don't. Determine contingencies with other projects. QA MUST have the final say over which projects are ready for release. If management does not support this you will be in bad shape fast!

Potential Issues:

- Release Criteria - Be firm with release criteria (bug count). If you aren't, you will get walked on again and again and again.
- Hard Release Dates - Sometimes dates simply can't slide ...make sure you have a sufficient buffer in the schedule in case problems crop up ...because they will crop up.
- Large Projects - Typically these will exist in “other” QA environments, but will need to be rolled into the standard QA environments. It makes a lot of sense to roll in large components of these projects early for system testing (to verify no impact to the rest of the site and reduce the size of the release later).

Client Side Testing

- Operating Systems
- Browsers
- Connectivity

It is critical to have a good client side test lab:

Operating Systems - Ensure MS NT and Windows, Mac and possibly others.

Browsers - IE, Netscape, Mac, AOL, WebTV (software simulator works pretty well) ...possibly Opera (closest to HTML 4.0 standard), etc.

Connectivity - Ensure different modem speeds to verify performance. Ensure at least one out of state provider (AOL is out of Virginia).

Try to figure out what your customer will use based on statistical analysis.

Do not let the customer be your tester!

Client Side Testing - New Technology

- CSS (vs. version 3.X browsers...)
- HTML 4.0 standard (versus browser standards)
- JavaScript (versus ASP)
- WebTV
- Palm/Cell Phones ...what's next?

CSS - Cascading Style Sheets - Work like include files, but provide a central location in which to change font sizes and other HTML characteristics. If you use CSS, Version 3.0 browsers and WebTV will no longer will work. This needs to be handled in a graceful manner with an error screen or redirect to non-CSS pages.

HTML 4.0 - The standard that wasn't a standard ...Netscape and IE follow much of this standard, but not all of it. As a company it should be decided whether this standard should be adopted as a whole ...or if the focus is simply to verify functionality (in all browsers, which takes longer to verify).

JavaScript - Works well is most browsers, but not so well in earlier versions. Later versions recognize more and more JavaScript. Find out your target audience and determine before design which version of JavaScript to adopt and how this will impact your user group.

WebTV - The latest is similar to Version 3.x browsers ...but this audience is growing...

Palm / Cell Phones next ...the point is that we will see more and more platforms in the future and testing considerations will need to be thought through...

Tools

- Bug Tracking Software
- Load/Stress Testing
- Automated Regression Testing
- Database Testing
- Visual Source Safe

Bug Tracking Software - This purchase is the most important one you will make - Pick a good tool which provides the flexibility to determine the data submitted, as well as bug tracking process. Ensure solid queries and reports, but also web access for remote availability.

Load/Stress Testing - Load and Stress testing is focussed on removing risk. Evaluate the tools on the market to match your needs, but most importantly ensure you have good staff who know how to accomplish this task.

Regression Testing - The primary goal behind regression testing is to reduce tester burnout, reduce resource needs and improve accuracy or results. The payback is the 10th time these scripts are run they have paid for themselves. Focus on less dynamic parts of the site which require significant regression testing for each release (such as transactions, search, etc.).

Database testing - SQL Client has worked fine as long as the permissions in the test environment were set up correctly.

Visual Source Safe - We use this primarily to version control test plans and other important testing documents.

Processes

- Projects
- Code Promotion
- Bug Tracking
- Testplans

There are many different processes which touch QA. The most critical are:

Projects - Schedules, milestones, release criteria and so on... QA needs to be involved at the earliest stages to help determine adequate testing time and staffing (otherwise quality will suffer).

Code Promotion - Since code moves through various QA environments on the way to production it is critical that QA have insight into this process.

Bug Tracking - This is the process that QA needs to own and define.

Test-plans - Really part of the project lifecycle, but there should be templates created and expectations set for this process within QA.

Processes - Projects

- Assignments
- Milestones and Documentation
- Promotion Criteria
 - Promotion through environments
 - Red Flags
 - Decision to release

Project Assignments are important. Skills sets need to be matched correctly, complexity also varies greatly and finally testers will do better on those projects they are interested in.

Milestones and Documentation - This is usually driven by the Project Manager, but projects should not be tested if documentation has not been delivered. This includes the Project Overview, System Design, Design Templates, Build Documents and possibly many others depending on the nature and complexity of the project. These are critical in order to form a testplan. Demand them and shut down a project until you get them.

Promotion Criteria - The different QA environments can be used as gates ...criteria set at each gate to pass through to the next level. Missing docs or late projects can be used as an early warning system or sorts to have a sense of which projects may make release and which will not.

Processes - Bug Tracking

- Setting Priorities
- Key Checkpoints
 - Submitter
 - Test Lead
 - Project Lead
 - Developer
 - Submitter
- Bug Triage Meetings (break “log jams”)
- Production Bugs (easy to ignore after release)

Setting Priorities - It is somewhat controversial whether QA sets priorities or the PM. It determines developer resource allocation, so we have allowed PMs to set priorities. These are clearly defined however, so violators can be counseled in the error of their ways.

Key checkpoints - The main checkpoints are that the PM sets priority and ONLY the submitter or a QA person can close a bug. This should be easily set up in the permissions in good bug tracking software.

Production Bugs - The problem when you don't have zero tolerance for bugs is that they make it to production ...and then the PM and developer are on to their next project. The bug automatically loses priority. We have assigned a PM to track these bugs and assign resources to close them as a separate project ...but the verdict is still out as to whether we should tighten release criteria (and possibly seriously slow releases down for lower priority bugs).

Processes - Testplans

- Version Control
- Template
- Content
- Migrating to automation

Good Testplanning is of course essential for a solid QA effort. A good template is necessary as well as version control.

They should be written so as to be reusable by anyone somewhat familiar with the website and basic testing methodology.

They can later be used as a basis for regression testing using automation tools.

Finally, one idea I had was to publish a testplan for a company-wide “bug bash” in HTML. The starting URL was listed for each area tested as well as starting times. Since everyone was literally “on the same page” at the same time, it provided a non-automated load test as well as a fairly simple testplan for a layperson to follow.

Additional Considerations

- N-Tiered Test Environments
- Dynamic (web) versus Static (shrink-wrap) product
- Life-critical Content
- eCommerce Considerations
- Data Security

N-Tiered - eCommerce Internet sites are complicated ...the test environments are complicated. We usually have 6 to 10 servers in any given test environment and then we pass data to test systems at the fulfillment center in Ohio. It is essential to have a good group in CM and IT to maintain this ...it is simply beyond QA to maintain this. It also means these environments can easily be corrupted or get out of synch with one another.

Dynamic versus Static - We have done the full circle from mainframe to client to web. Changes can be done in one place again and have global impact. This provides greater flexibility and speed to market, but also severe consequences for mistakes. Keep releases manageable and maintain good criteria for releases.

Life Critical Content - If there is any life critical content on your website ...ensure the right people are looking at it as part of the QA process (at CVS.com this is typically the pharmacists). Do not EVER own the content, but ensure it is properly reviewed.

eCommerce Considerations - People can get rather annoyed if someone starts charging on their credit card. Or someone offers to sell their spending habits to a junk mail company. CVS.com is a member of TRUSTe and XXXXX which means that we follow stringent criteria with regard to these and other related things. These are very worthy nonprofit organizations and should be considered by your organization.

Data Security - Peoples sense of security is paramount and data needs to be carefully guarded. This needs to be thoroughly tested with it's own focus. Outside experts are recommended.

Must Do List

- QA Must Own the Release Decision ...and Executive Management must agree to this.
- Set Release Criteria and stick to it!
- Hire (and maintain) a good QA staff!
- Select a good Bug Tracker and keep on top of bugs.
- Only allow QA or Submitter to close bugs.
- Do not own the success or failure of a project ...only own the quality!
- Use Risk/Benefit Analysis where ever possible.
- Assign QA Resources well.

This is a list of things that **MUST** happen if you are to be successful!

Walk through the list...

About CVS/CVS.com

America's #1 Pharmacy: A Marriage of Clicks and Bricks

- 4,100 Stores in 26 states and the District of Columbia
- 20 specialty pharmacy stores nationwide
- 12,000 pharmacists
- 281 million prescriptions filled in 1999
- \$18.1 billion in sales in 1999
- 9,000+ insurance plans
- 55 million customers
- 2.5 million customers served per day
- Seattle, Ohio, Rhode Island

- Launched in January 1999 as soma.com
- Acquired by CVS/pharmacy in June 1999
- Site relaunched as CVS.com in August 1999
- Site enhanced and relaunched in March 2000
- CVS.com is the world's first Internet pharmacy, the only online pharmacy to be fully integrated with an offline retailer. This unique relationship allows CVS.com to provide its customers with the best of both on and offline worlds: savings, ultimate convenience, and tremendous product selection
- We see the Internet as a logical and natural extension of our overall convenience and value strategy.
- CVS is the largest pharmacy chain in the nation in terms of stores
- CVS is the largest provider of prescription drugs in the nation with 281M filled last year for 55M customers
- CVS Procure launched in June 1999. Procure is committed to the complex and expensive drug therapies required by patients with chronic diseases and conditions. These include HIV/AIDS, organ transplants, infertility and other conditions. Goal is to have 1000 Procure pharmacies nationwide within four years, enabling CVS to cover the majority of the targeted patient population.

Thank You

www.CVS.com

Contact Information: JamesH@email.cvs.com

Thank you very much for your time!