Paper 007-2009

Visual Guide to SAS® Web Application Development

Sy Truong, Meta-Xceed, Inc, San Jose, CA

ABSTRACT

The browser wars have entered a new era with Google Chrome challenging Microsoft Internet Explorer's dominance. One of the driving forces and strategies for Google Chrome is to have the browser function as a platform to deliver interactive applications rather than just static websites. The new browser is designed from the bottom up with the purpose of optimizing performance for delivering applications. SAS® software has traditionally been reserved for power users performing analytics for specific vertical business intelligence needs. The browser war illustrates the maturity of the delivery of web applications. This presents opportunities for SAS solutions to be delivered to a wider audience with minimal user requirements outside of a web browser. This paper is a visual guide describing the steps needed to efficiently develop and deploy user friendly web applications with powerful server side SAS processing.

INTRODUCTION

A picture is worth a thousand words. This paper employs this "web" philosophy by presenting technical concepts for web application development through the use of visual screenshots and diagrams. It presents methods to optimize the delivery of information in a unique and compelling way. The visual methods of this paper are analogous to the content of the paper which is to optimize the delivery of user friendly software applications. The visual approach is also intended to be used quickly upon review, rather than having to read the text in great detail as in the more traditional paper with long descriptive text. The topics will be presented in this thumbnail view which functions similarly to a table of contents.

Task 1
Project Definition

Task 2 – User Requirements and Functional Specifications

Task 3
GUI Prototype Design

Task 4 – Usability Testing

Task 5 – Browser Considerations

Task 6 – Design Considerations

Task 7
JavaScript & AJAX Implementation

Task 8 – Server Side Logic

Task 9 – Application Management

Task 10 – Middleware Performance Tuning

Task 11 – Client Side Optimization

Task 12 – Context Sensitive Help

Task 13 – Validation and Testing

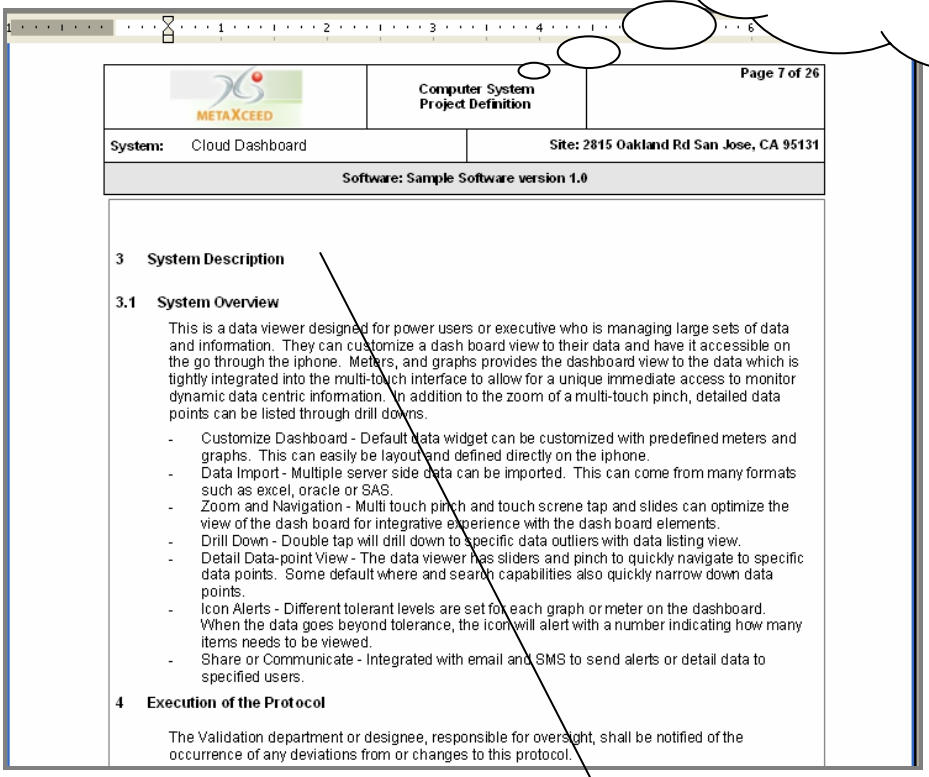
Task 14 – User Documentation & Tutorials

Task 15 – Portal Deployment

Task 16 – Discussion Forum Support

TASK 1 – PROJECT DEFINITION

Application "Mission Statement"
A clear and concise definition is an essential first step towards guiding all aspects of development.



METAXCEED	Computer System Project Definition	Page 7 of 26
System: Cloud Dashboard		Site: 2815 Oakland Rd San Jose, CA 95131
Software: Sample Software version 1.0		

3 System Description

3.1 System Overview

This is a data viewer designed for power users or executive who is managing large sets of data and information. They can customize a dash board view to their data and have it accessible on the go through the iphone. Meters, and graphs provides the dashboard view to the data which is tightly integrated into the multi-touch interface to allow for a unique immediate access to monitor dynamic data centric information. In addition to the zoom of a multi-touch pinch, detailed data points can be listed through drill downs.

- Customize Dashboard - Default data widget can be customized with predefined meters and graphs. This can easily be layout and defined directly on the iphone.
- Data Import - Multiple server side data can be imported. This can come from many formats such as excel, oracle or SAS.
- Zoom and Navigation - Multi touch pinch and touch screne tap and slides can optimize the view of the dash board for integrative experience with the dash board elements.
- Drill Down - Double tap will drill down to specific data outliers with data listing view.
- Detail Data-point View - The data viewer has sliders and pinch to quickly navigate to specific data points. Some default where and search capabilities also quickly narrow down data points.
- Icon Alerts - Different tolerant levels are set for each graph or meter on the dashboard. When the data goes beyond tolerance, the icon will alert with a number indicating how many items needs to be viewed.
- Share or Communicate - Integrated with email and SMS to send alerts or detail data to specified users.

4 Execution of the Protocol

The Validation department or designee, responsible for oversight, shall be notified of the occurrence of any deviations from or changes to this protocol.

EXECUTIVE LEVEL DEFINITION

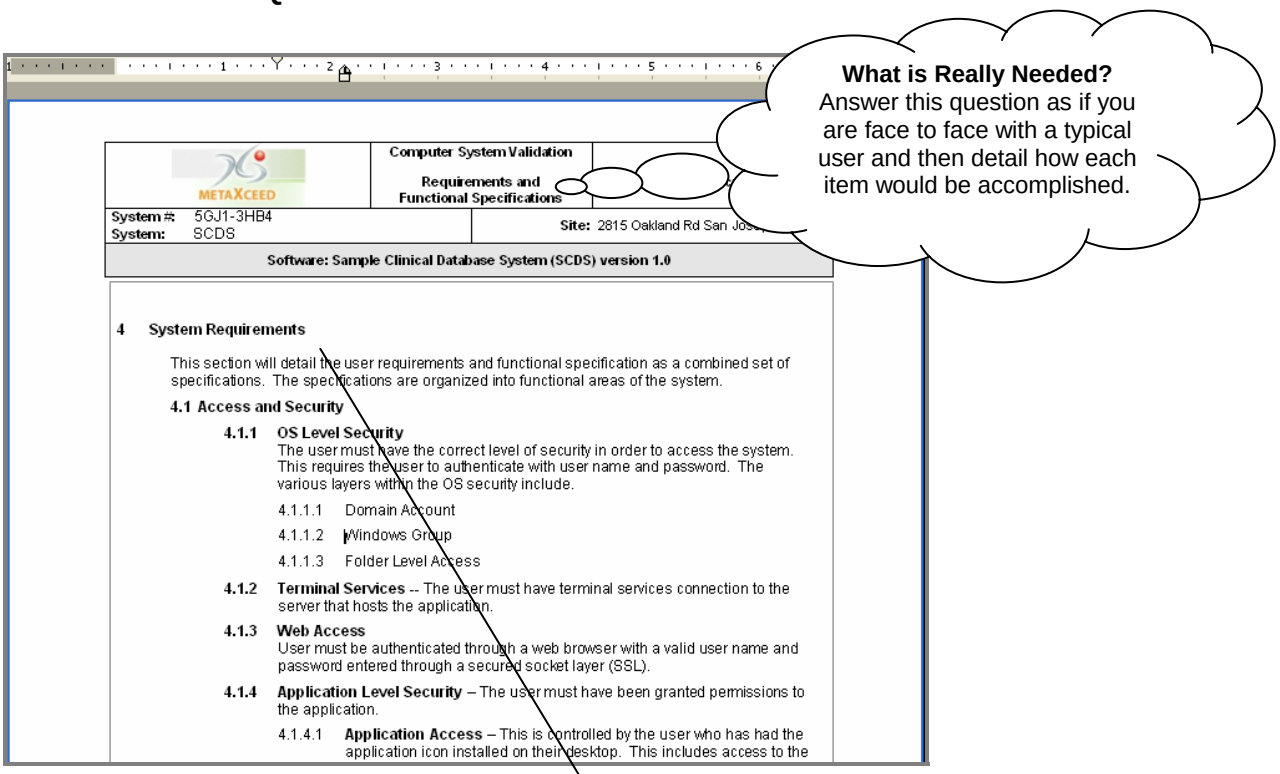
What it is – Define in concise description what is the core purpose and function of the system.

Limit Scope – For the initial release, limit the definition to the bare essentials. Leave extra features for future releases.

Function Before Technology – Focus on defining what problem is being solved first before defining how or what technology will be used to solve the problem.

Leverage Platform – Understand communication advantages of web applications and ability to be delivered ubiquitously. Take advantage of these inherent benefits during definition.

TASK 2 –USER REQUIREMENTS AND FUNCTIONAL SPECIFICATIONS



ENUMERATE ESSENTIAL WISH LIST

Your Wish is My Command – Interview typical users with Genie like courtesy and document all wish list items.

Adherence to Definition – Stay within the project definition and scope of the original set of objectives.

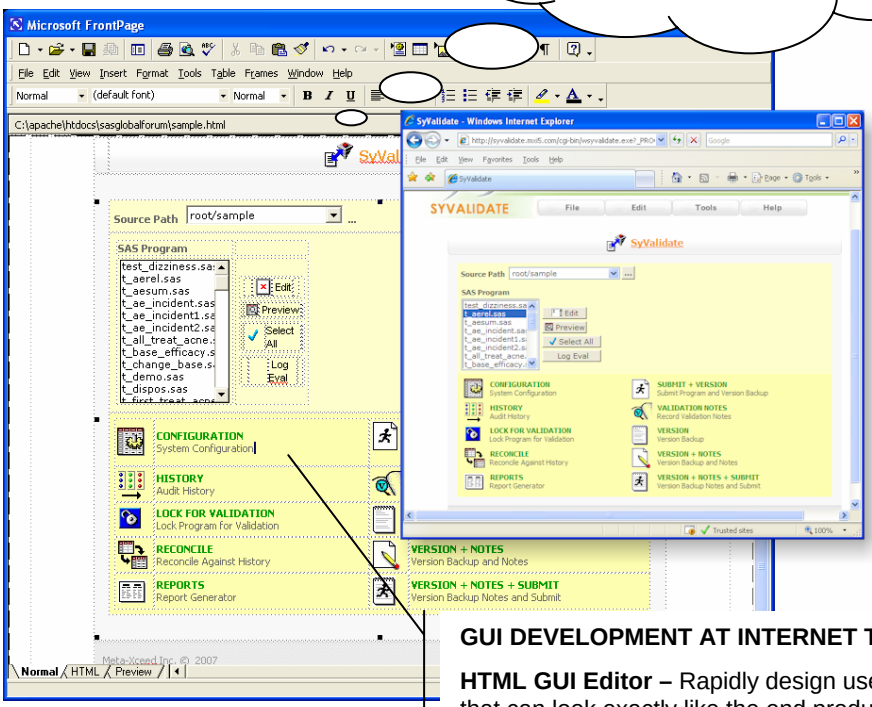
Devil in the Details – Document every detail of each user requirement deciphering each individual component. Split each requirement as granular as you can before any development.

No Coding – At this stage, things can change dramatically so avoid development or even prototyping until all specifications are finalized.

Requirement to Functionality – For every requirement, there is at least one functional specification. This will detail the answer to every question that is posed by the requirement. Document for the purpose of guiding development.

TASK 3 – GUI PROTOTYPE DESIGN

Pictures Worth a Thousand Words
Prototype with HTML and CSS efficiently without lengthy backend logic coding.



GUI DEVELOPMENT AT INTERNET TIME

HTML GUI Editor – Rapidly design user interfaces that can look exactly like the end product as easily as editing with a word processor.

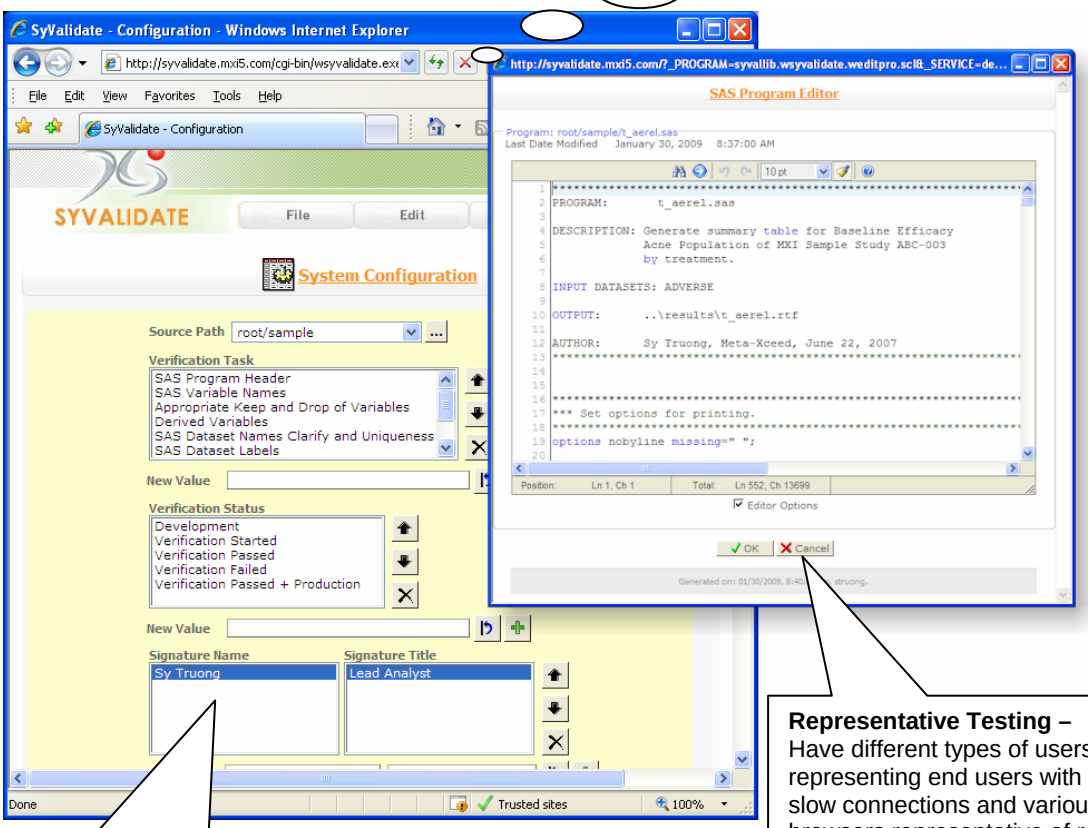
Consistent CSS – Place all font and color decisions into CSS. This will handle display element standards and modularize deployment.

Instant User Feedback – Place prototype on an intranet or secured internet site so users can review instantly. Comments and feedback can be updated in GUI in an expedited manner.

Separate GUI and Backend – Database design and business rules logic are developed in conjunction but can be separated. This allows for rapid GUI parallel development with distinct different skills from web designers that are not necessarily database or SAS programmers.

TASK 4 – USABILITY TESTING

It's all About the Users
Usability correlates directly with effectiveness of systems. Early user testing makes all the difference.



Early Testing – Users can test fictitious data from HTML form GUI before backend code is complete.

Testing Log – User interaction can be captured by the application but also web server logs are useful.

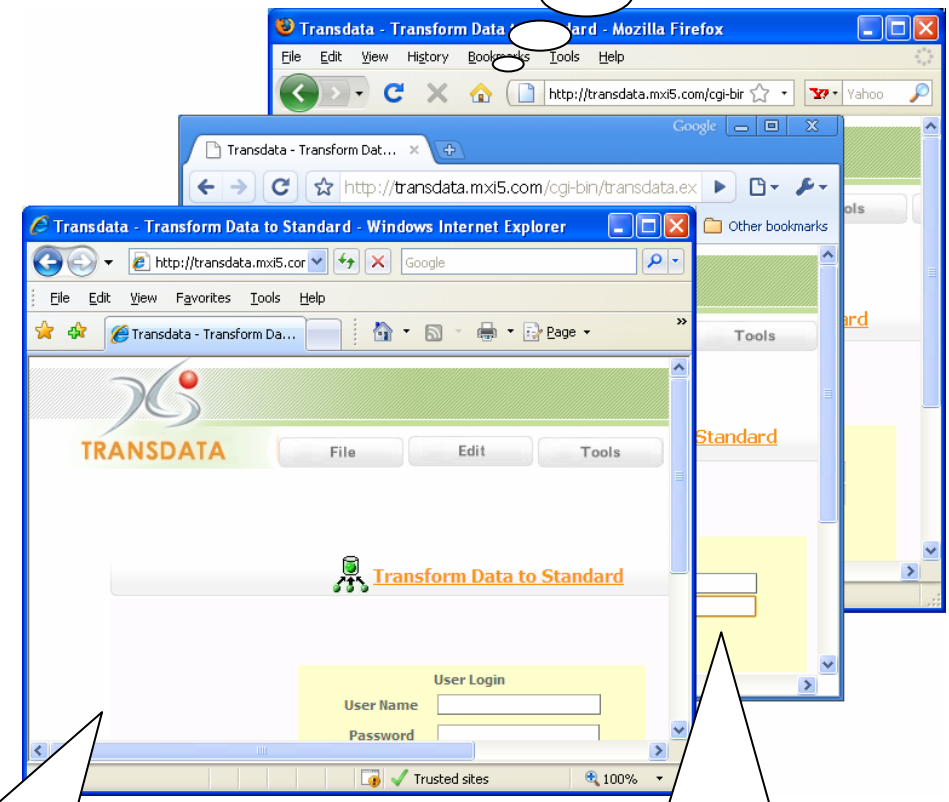
Employ Social Technologies – Have users comment upon testing beyond email with blogs, forums and wikis.

Representative Testing – Have different types of users representing end users with slow connections and various browsers representative of real world deployments.

Global Users – Multiple simultaneous testing can occur easily with users testing from anywhere in the world with internet connection.

TASK 5 – BROWSER CONSIDERATIONS

Target Browser Audience
An evaluation of an intranet or extranet release of the software will determine the target browser.



WEB BROWSER SELECTION

User's Install Base – Users may already have desired web browser installed. Develop for largest existing install base.

Intranet vs. Extranet – Intranet standards and software restrictions can limit to specific browser code and methodologies. Release to extranet only after consideration for extra security and support are understood.

RESOURCING DEVELOPMENT

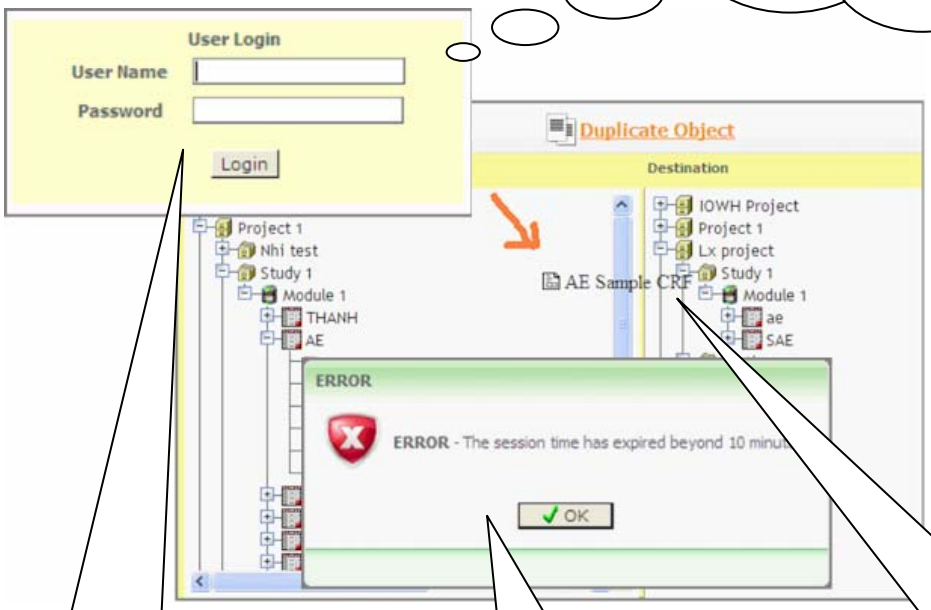
Application Performance – Processing of AJAX of JavaScript performs differently among browsers. Devote additional resources for this tuning.

Scripting Support – Some JavaScript are not supported on certain browsers. Allow for additional development efforts for different scripts for each browser.

Development Resources – Are there enough resources to support all browsers? Each new browser includes a new set of coding and validation testing.

TASK 6 – DESIGN CONSIDERATIONS

Distinctive Web Platform
Web applications operate very differently from client server or desktop applications. Develop solutions to handle these differences.



Authentication – Web users are anonymous and need to log in using unique ID and passwords.

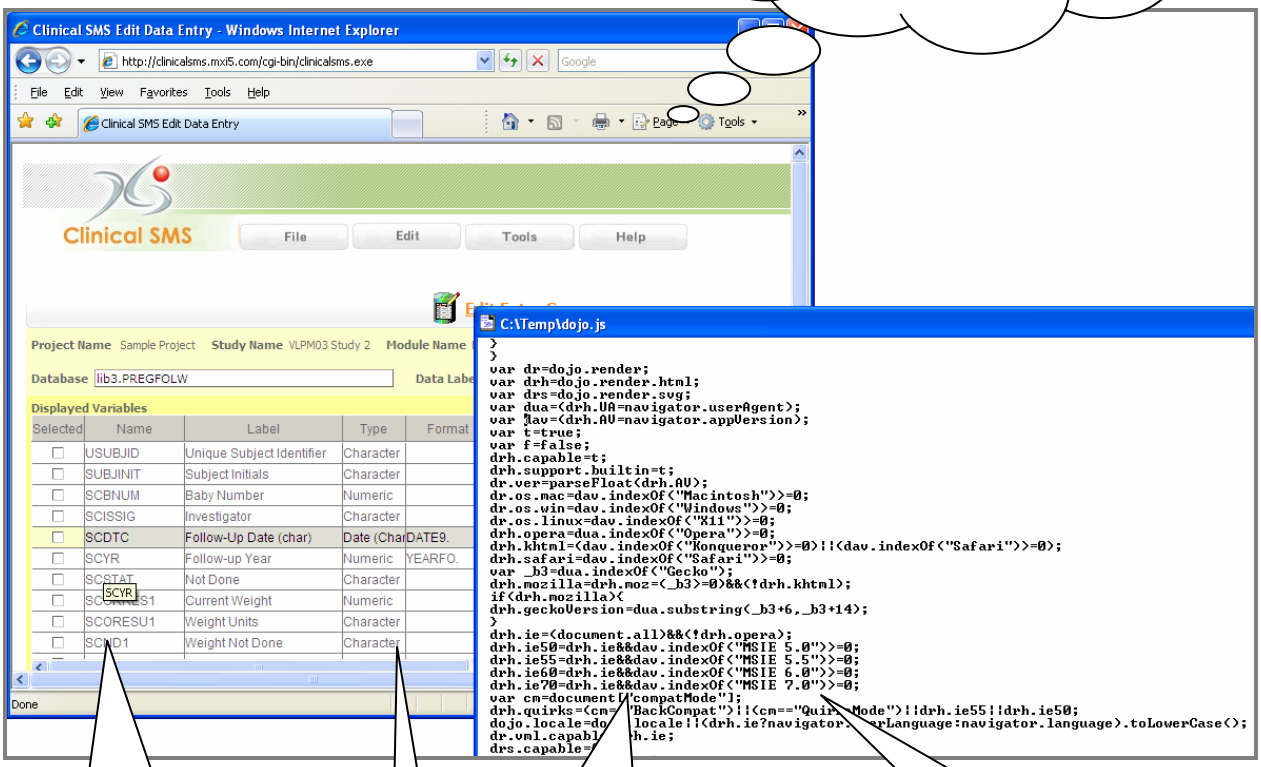
Session Management – Users connect and disconnect to the server at any time and it is considered stateless. Management of the user's state knowing when they are logged on and what screen they are at is required.

Client vs. Server Processing – Client scripts increase performance on small tasks while server side power of SAS is needed. The requirement on user client machine is kept to a minimum for wider access.

GUI Capabilities – Drag and drop and other interactive elements such as explorer tree view are now available with many JavaScript and AJAX libraries. This was once only available on desktop applications. Without this limitation, web applications can be implemented for wider audiences.

TASK 7 – JAVASCRIPT AND AJAX IMPLEMENTATION

Enhance Interactivity of Client
Web applications can have greater interactivity when combined with JavaScript, AJAX and XML.



Ajax Tools – AJAX libraries can be utilized to accomplish interactivity such as spreadsheet view with cell manipulation.

XML Data – Data can be transferred in XML format and rendered in the browser to be presented in many formats including spreadsheet view.

Browser Support – JavaScript can have different functions for each browser. Have a set of scripts for each browser that is supported. This includes different versions of each browser.

Client Optimization – Scripting for each client browser and version can be optimized for each browser. Different script logic for each version is customized for compatibility and performance.

TASK 8 – SERVER SIDE LOGIC

Multiple Language Support
Server software needs the flexibility for handling multiple languages to deliver full range of applications.

BUILD: SOURCE TEMPLIB.CLINSMS.TEST.SCL (E)

```
00001     *** Generate HTML header ***;
00002     submit;
00003 <script type="text/javascript" src="&jspath/dojo/dojo.js"></script>
00004 <script type="text/javascript">
00005         dojo.require("dojo.widget.Dialog");
00006 </script>
00007
00008 <script language="JavaScript" src="&jspath/result_error_status.js"></script>
00009 <script language="JavaScript" src="&jspath/about_dialog.js"></script>
00010 <script language="JavaScript" src="&jspath/info_dialog.js"></script>
00011 <script language="Javascript" src="&jspath/waiting_dialog.js"></script>
00012
00013     endsubmit;
00014
00015     call method("winfo_dialog_methods.scl","WriteDialogMarkup");
00016     call method("wabout_dialog_methods.scl","WriteDialogMarkup");
00017     call method("wpreviewdat_methods.scl","WriteJavaScript");
00018
00019     cversion = symgetn('sysver');
00020     if cversion < 9.1 then do;
00021
00022        *** Create the ODS Style Sheets for all Reports ***;
00023        submit continue;
00024  proc template;
00025     define style Styles.MXISTYLE;
00026        parent = Styles.Default;
00027        style Default /
00028           leftmargin = 0
00029           background image = _undef_
00030           cellpadding = 5
00031           rightmargin = 0
00032           watermark = off
00033           foreground = black
00034           cellspacing = 0
00035           bottommargin = 0
```

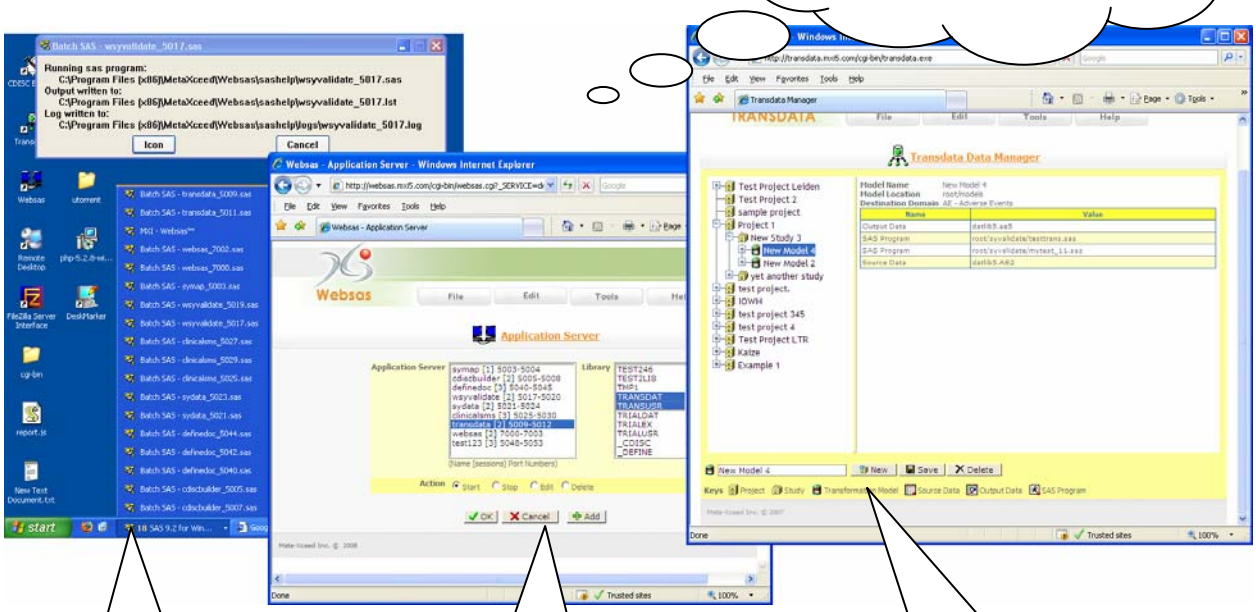
HTML, JavaScript and AJAX – The HTML and JavaScript can be dynamically generated on the server where it has access to databases and powerful server side tools. In this example, it is generated from within a SAS SCL program.

SAS SCL – SAS SCL can be the glue which ties all the other languages used. In this example, it can be used to generate other code including: JavaScript, Ajax, SAS Base. It also has direct hooks into SQL similar to PROC SQL and other engines making it the versatile foundation.

SAS Base – SAS Base or Foundation SAS is very extensive and can encompass many SAS PROC such as the ODS PROC TEMPLATE in this example. It can call routines from all SAS modules to handle database access such as in PROC SQL or powerful analytics such as in SAS/STAT. The power of SAS coupled with powerful hardware on the server makes it a scalable solution to handle any dynamic demand.

TASK 9 – APPLICATION MANAGEMENT

Managing Dynamic Servers
Each application can have multiple servers that change depending on usage. Careful management is essential for optimal performance.



Multiple Application Servers – Multiple applications can be served on the server by separate SAS sessions. Each SAS session communicates through a separate TCP Socket port for optimal communication. The multithreading of each SAS session can fully utilize load balancing. Additional machines can be added to handle more applications to scale to user demand.

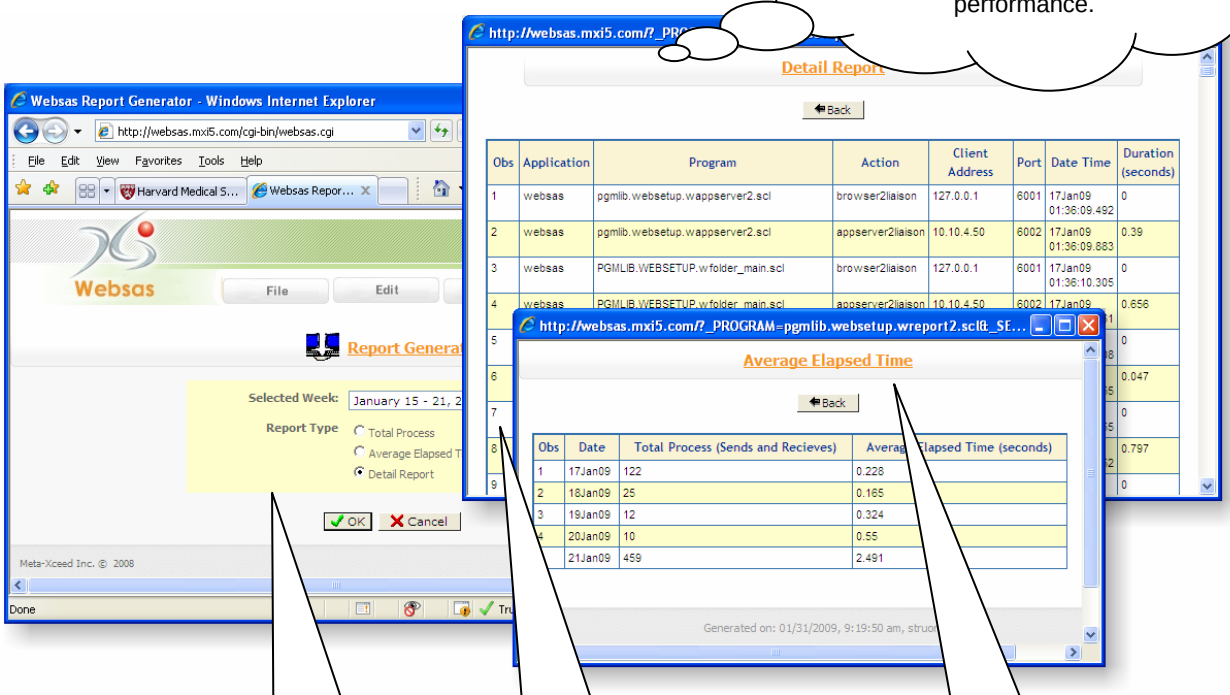
Manage Server Centrally
All applications can be managed through a single interface that is also web based as in the example of Websas™. The main task of management includes the following tasks:

- Starting and Stopping applications servers dynamically.
- Assign different communication socket ports to each application.
- Generate report detailing each application performance used for optimization tuning.

Delivered Application
Application is accessed through one URL address named after the application for ease of access. This can also be a link from existing portals for expedited navigation.

TASK 10 – MIDDLEWARE PERFORMANCE TUNING

Tuning Requires Monitoring
Resource demands fluctuate depending on usage and size of data. Monitoring and adjusting is required for optimal performance.



Dynamic Server Status
An administrator needs to be able to review the server status dynamically at any point. An effective method is to deliver this as web reports.

Dynamically Add Servers
Upon review of status, the administrator needs to be able to dynamically add or remove application servers. This will allow for immediate real time control. This is accomplished by adding or subtracting servers which are SAS sessions communicating on distinct socket ports.

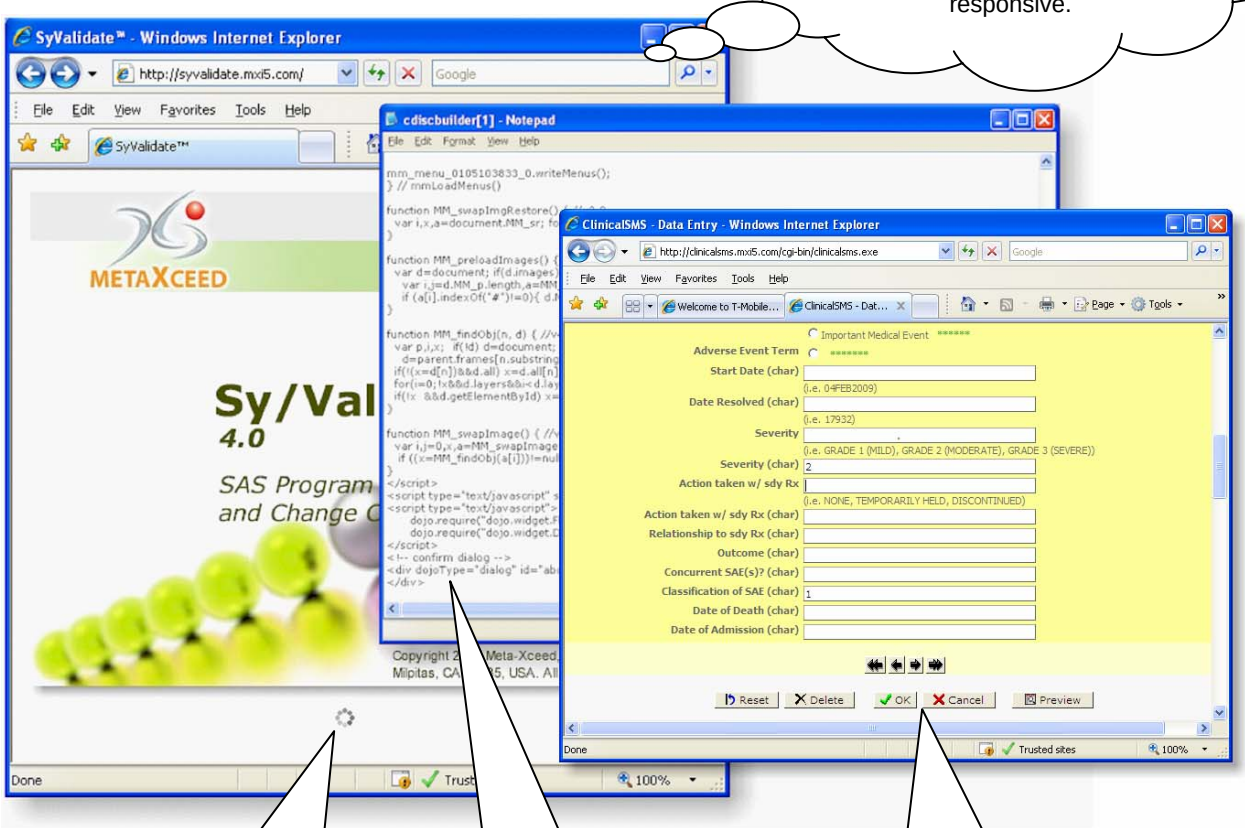
Auto Recovery
The application server can restart automatically if it crashes for consistent and continuous uptime.

Historic Log
A complete log can be viewed weekly since the entire log can be lengthy. Items in the log can also be searched to pin point errors. All log information can be presented to users and administrators through the web browser.

Aggregate Performance
A summary of performance averaged out by day will provide an administrator with a long term view. This is needed to make adjustments and tune server according to shifting resource demands.

TASK 11 – CLIENT SIDE OPTIMIZATION

Client Browser
The client browser interacts directly with the user. Optimization makes the experience much more responsive.



Splash Cache Load
All common image files are preloaded during the display of the splash screen into cache. Subsequent screens will load and display from cache much faster.

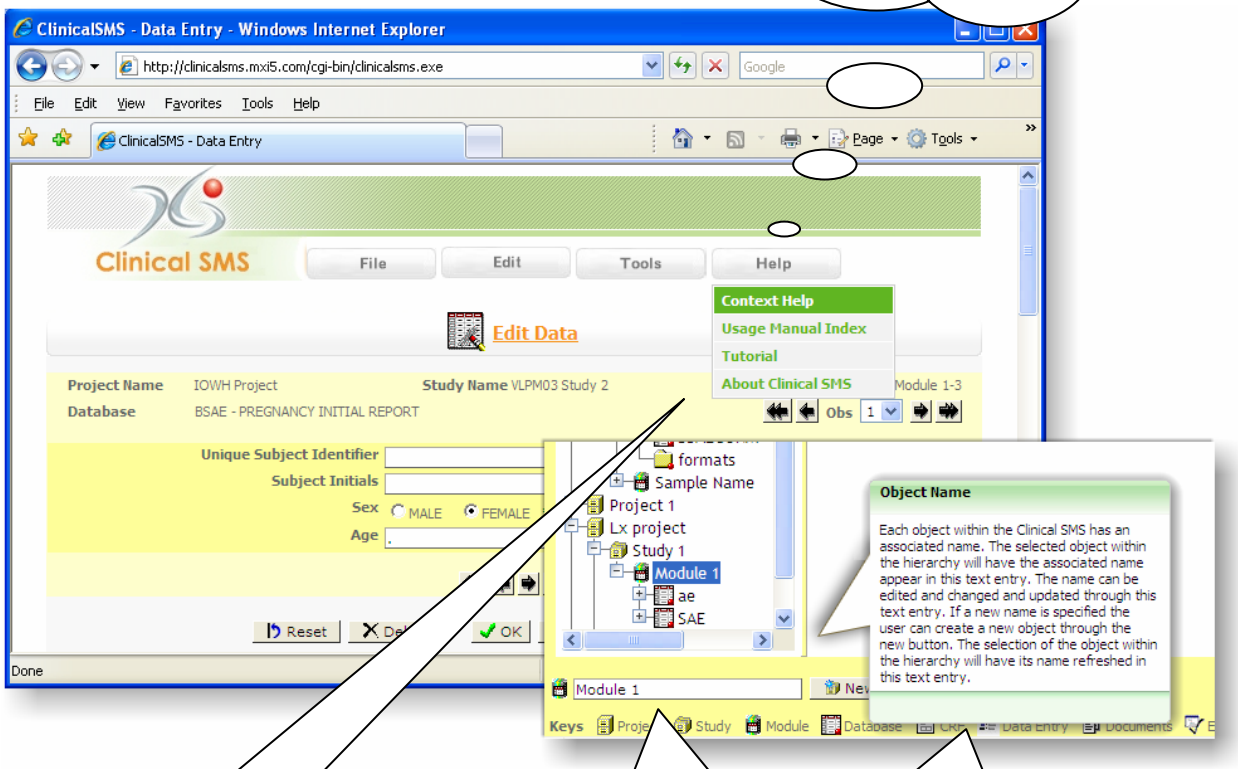
Script Bottom Load
Length scripts and arrays are added to the bottom of HTML files. This allows for browsers to render the display of the screen first before loading the logic of the scripts for increased responsiveness.

Form Error Checking
For valid values on text entries and other form elements can be performed on the browser. This saves a trip back to the server for immediate response to user.

Wizards
For a long sequential set of forms, a step by step wizard can be applied so each form on the client does not get too large and confusing for user.

TASK 12 – CONTEXT SENSITIVE HELP

When You Need Help
Help information is useful when it is within the context of the screen that the user is in.



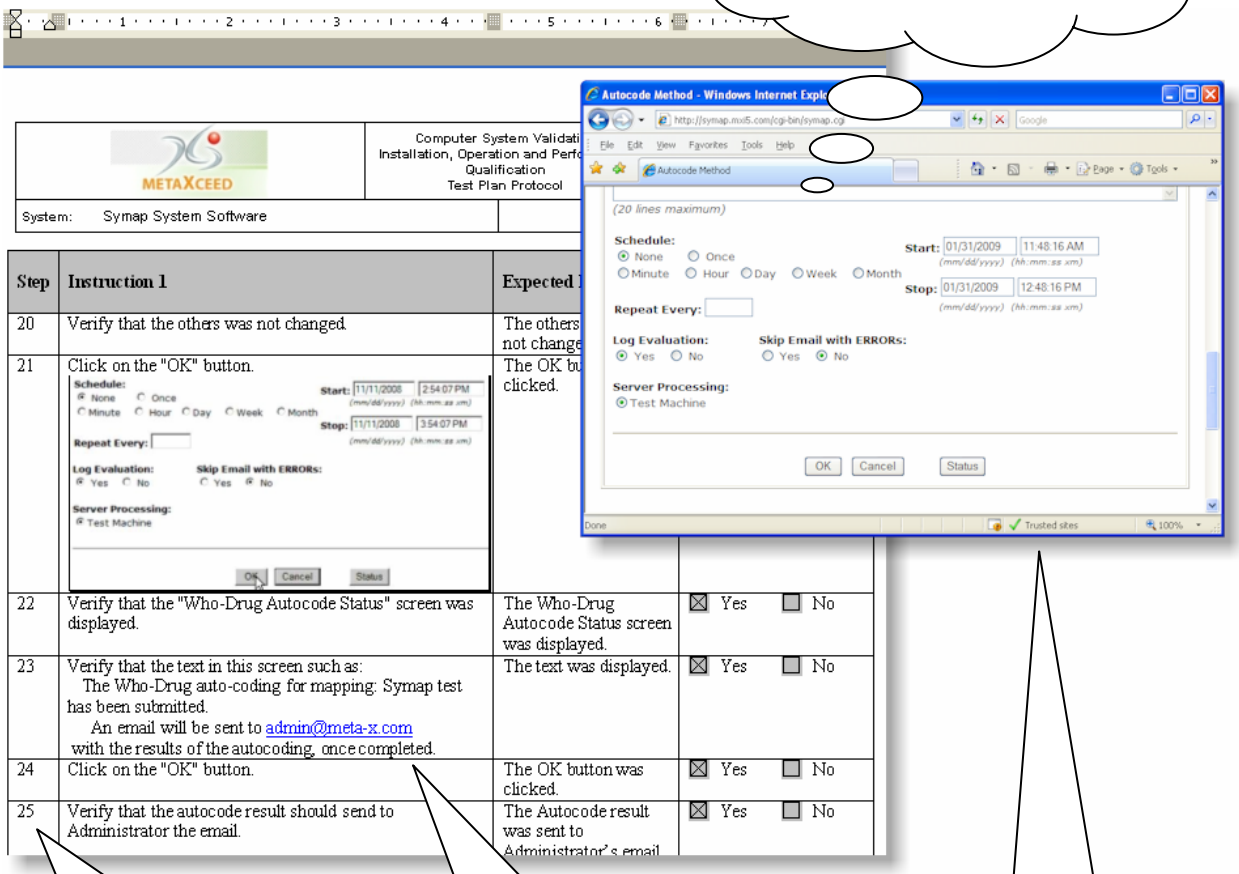
Help Access
Access to help information including user manuals and tutorials is available on every screen. An effective method is through a menu. The use of a quick key such as F1 is also expedient.

Video Tutorials
Flash video tutorials can be more effective than static text manuals in many cases. This includes audio and interactive examples. Depending on the user and material, this can prove to be most effective.

Popup Help
An explanation of what each object is on the screen is sometimes enough to clarify the user's needs. This can be accomplished with a mouse over pop-up. The bubble would temporarily provide more contextual information that is not obvious.

TASK 13 – VALIDATION AND TESTING

User Friendly Testing
System testing can be mundane and prone to errors. User friendly instructions with screenshots are helpful.



METAXCEED	Computer System Validati Installation, Operation and Perf Qualification Test Plan Protocol
System: Symap System Software	

Step	Instruction 1	Expected	
20	Verify that the others was not changed.	The others not change	
21	Click on the "OK" button.	The OK bu clicked.	
22	Verify that the "Who-Drug Autocode Status" screen was displayed.	The Who-Drug Autocode Status screen was displayed.	☒ Yes ☐ No
23	Verify that the text in this screen such as: The Who-Drug auto-coding for mapping: Symap test has been submitted. An email will be sent to admin@meta-x.com with the results of the autocoding, once completed.	The text was displayed.	☒ Yes ☐ No
24	Click on the "OK" button.	The OK button was clicked.	☒ Yes ☐ No
25	Verify that the autocode result should send to Administrator the email.	The Autocode result was sent to Administrator's email	☒ Yes ☐ No

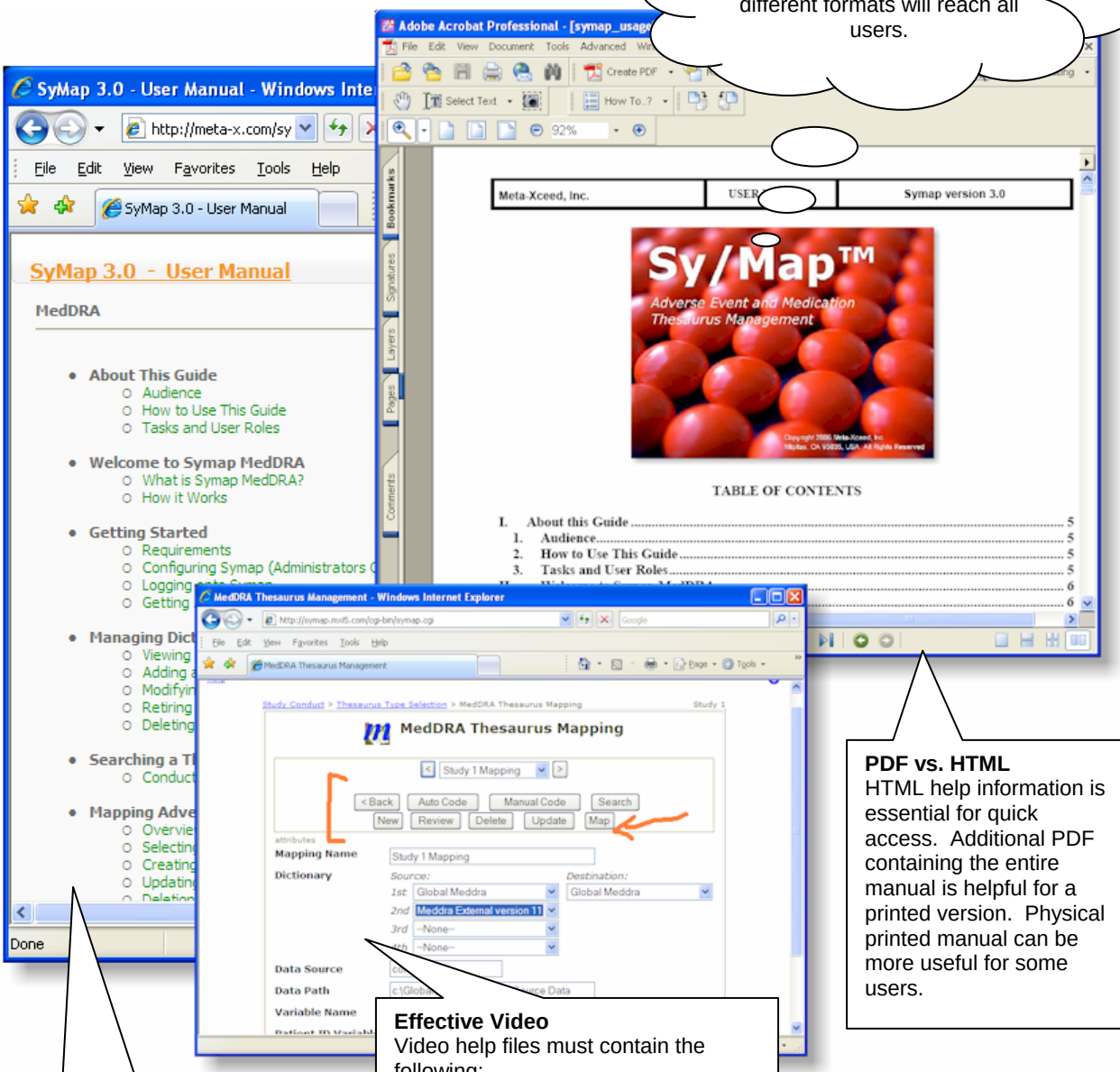
Validate According to Risk
Formal validation test scripts such as this are only required for critical path applications that affect many aspects of your work. Evaluate the risk on a tool and apply the right amount of validation according to the risk.

Effective Test Plan
A test plan includes test scripts with clear instructions on how to test and expected results for success. All test cases are developed to fulfill a system functional specification and requirement.

Visual Instructions
In addition to text instructions on the testing, screen shots are useful for instructing how the test should be done. Screen shots of the results are also useful in testing documenting results.

TASK 14 – USER DOCUMENTATION & TUTORIALS

Users Learn Differently
Different users learn in different ways. Multiple help files in different formats will reach all users.



PDF vs. HTML
HTML help information is essential for quick access. Additional PDF containing the entire manual is helpful for a printed version. Physical printed manual can be more useful for some users.

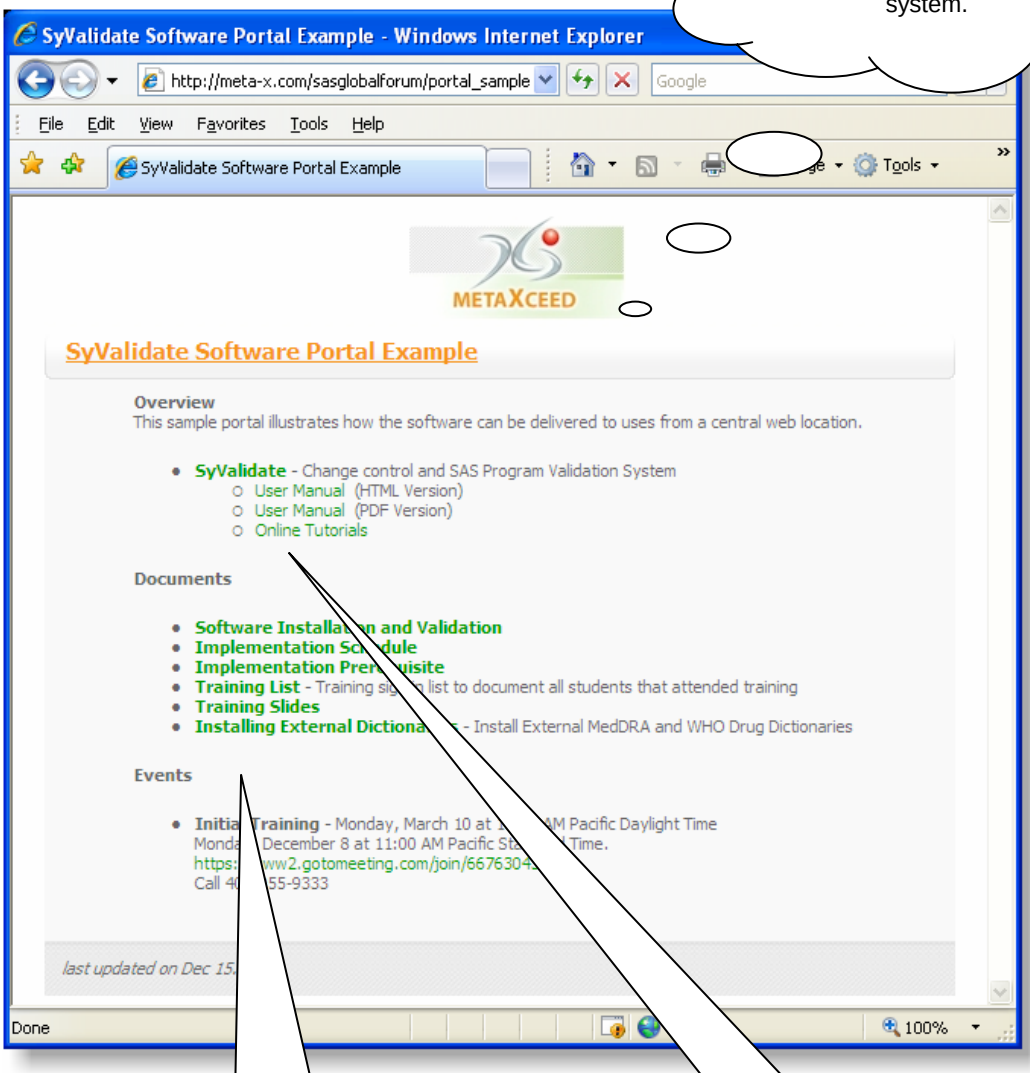
Navigational Manual
User manual table of contents HTML format is effective for ease of navigation. This integrates well into the system which is web based. Specific portions of the manual can be linked to certain pages within the application.

Effective Video
Video help files must contain the following:

- Flash format for optimal compression and fast delivery.
- Interactive display of the screen. Examples with drawing and pointers such as orange arrows clarify instruction.
- Concise audio instruction clarifying common usage examples of specific tasks.

TASK 15 – PORTAL DEPLOYMENT

Direct Traffic
Similar to a website, links from existing portal drives traffic toward a new application and system.

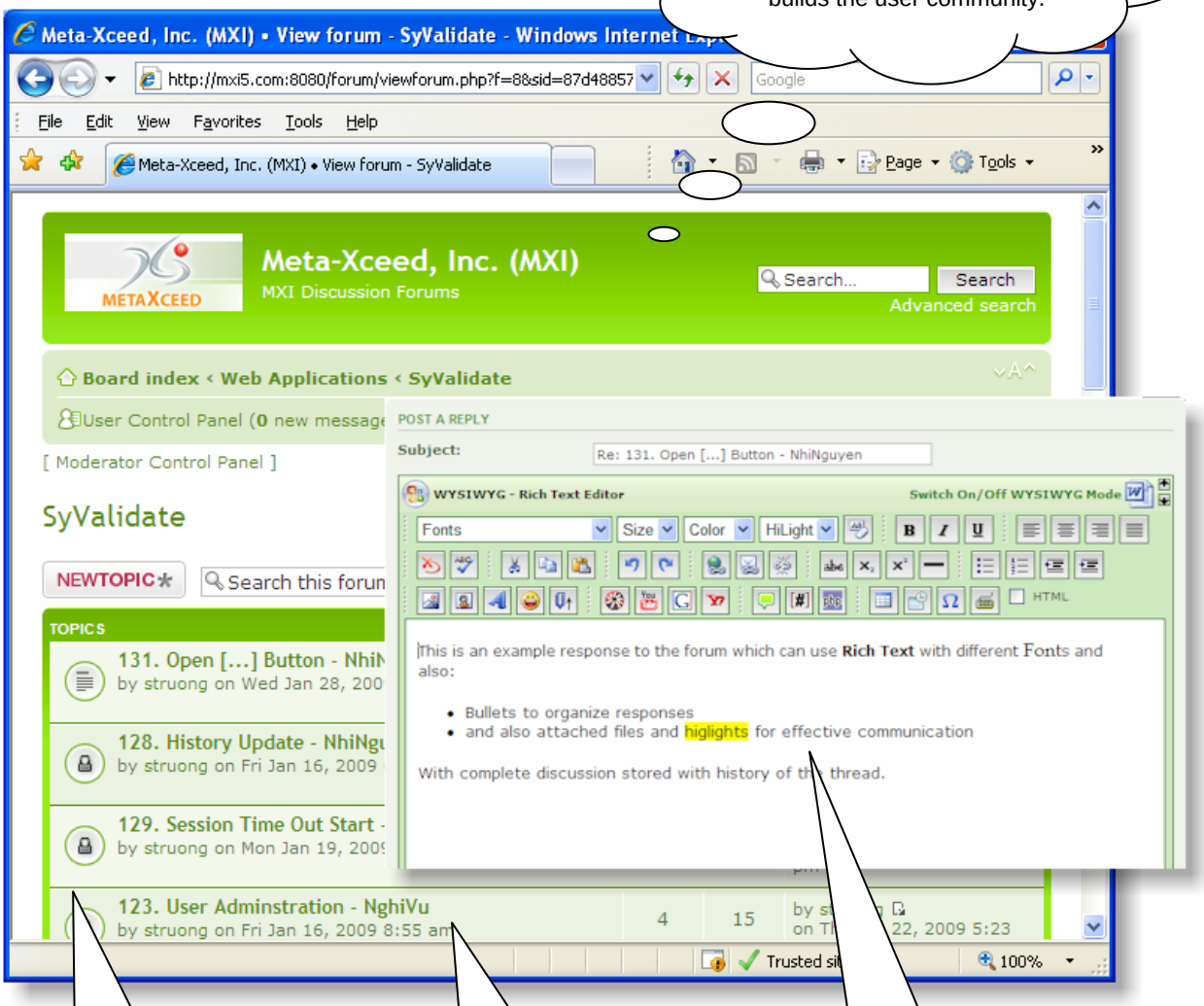


Good Place to Start
A home page for users can be a portal within an organization or department. Links to the system and related documentation is a good jumping point for users to enter the system.

Existing Portal
A link from within an existing portal is more effective than starting a new portal. This leverages existing users and directs usage to the application.

TASK 16 – DISCUSSION FORUM SUPPORT

Leverage Social Technology
Implement forum discussion in connection with application deployment for support and builds the user community.



User to User Help
Discussion Forum allows users to answer each others questions creating a more effective FAQ.

Moderate History
A well moderated support forum can filter out useless posts yet retain a complete audit trail of all user questions. This provides an organizational memory that is unmatched by other methods of support.

Effective Forum Posts
It is essential to clearly label responses with clear answers to questions. This includes: screenshots, text with organized bullets and use of highlights which can more effectively clarify issues posed on the forum. This example uses an open source phpBB forum tool.

CONCLUSION

The Web 2.0 is a revival for applications delivered through the internet. After the dot com bust, there was a perception that it would kill application delivered through the internet. Rather it was just a temporary speed bump as ecommerce websites and social network technologies have pioneered and continue to pave the way for web applications. SAS has been a compelling analytical tool for business applications and has adapted well to the computing environments of main frame computers to desktop and now the Internet. Web applications are uniquely different compared to their predecessor desktop software. Developing Web applications with SAS therefore requires a new approach. The new methodologies incorporate new web technologies such as AJAX on the client browser, Middleware and SAS on the server with XML as the data format transferred. The development process and deployment can also benefit from social networking technologies such as emails, blogs, forum and wikis. Implementing and deploying a web application has some similarities to a website such as accessing it through a website or linked through a portal. However, it is more sophisticated than a static website in that it has dynamic interactive objects including drag drop and videos with audio. Web technologies have matured and are setting the stage for a new platform which provides a new and more efficient way of delivering applications. Software is going through a fundamental shift from the likes of Microsoft delivering software in a box with a CD to a more dynamic website such as solutions from Google. SAS applications are currently well entrenched in large organizations and used by niche power users for business intelligence. In order for SAS applications to reach a larger audience, they must also make this transition and be effectively delivered as a web application.

CONTACT INFORMATION

Sy Truong is President of Meta-Xceed, Inc. They may be contacted at:

Sy Truong
2185 Oakland Rd
San Jose, CA 95131
408-955-9333
sy.truong@meta-x.com