

# SLOCs and Other Statistics

Adrian Kosmaczewski

2005-09-27

The Reuters' project I've previously written about is finally coming to an end, at least version 1.0. It is now "feature complete" and tomorrow it will be deployed in their own servers. That's a nice milestone!

Then I asked myself, how many lines of code have I written during this month? How many did I write per day in average? This is a unique opportunity to know, since I have been responsible for this little project from beginning to end, from analysis to design and implementation and testing, through the installation in their own infrastructure.

Just a quick reminder of the technical characteristics of the project:

- Intranet "Classic ASP" application, consisting of: 3 different screens, one big DHTML graphical editor (complex enough) with "WYSIWYG" capabilities, and a couple of minor pop-up windows. This application will be part of a larger project under the responsibility of other Reuters' teams, and provides a unique "viewing" capability for the managerial staff.
- Programming languages: VBScript, JavaScript, Transact-SQL, CSS
- Supported browsers: Internet Explorer 5.5 & 6, Mozilla Firefox (Mac and Windows) & Apple Safari (haven't tested in other browsers)
- SQL Server 2000 Database; the schema has been done by Reuters' staff, so I have just created the stored procedures, views and functions that the application uses to manipulate the data.

One important detail: this application has been completely written "by hand", from scratch; I have not used Visual Interdev to create it (I just don't want to use it) and it exploits VBScript and JavaScript object-oriented capabilities at its maximum... which seems like saying hard work for little benefit right? Well actually the idea behind this design principle is to give Reuters an application that would be easily portable to .NET in the future. "Classic" ASP is a dead technology that will no longer be supported by Microsoft in the future...

The different modules have been separated in layers following a classic 3-tier architecture, and I used the following open source utilities to help me build it:

- Classic ASP Framework by Christian Calderon;
- JavaScript Vector Graphics Library and

- JavaScript Drag & Drop Library both by Walter Zorn
- ASPUnit Unit Test Framework

Here's some statistics showing different checkpoints during the month of September; these have been done with Campwood Software SourceMonitor Version 2.0 (nice utility, quite awkward, but hey, it's free :)

Date	Lines (SLOC)	# of files	Cumulated working days	Average lines per day
5 Sep	1075	20	2	537.5
12 Sep	1437	25	5	287.4
15 Sep	2745	31	8	343.1
19 Sep	4379	38	10	437.9
22 Sep	5245	45	13	403.5
25 Sep	6070	56	15	404.7
26 Sep	6235	58	16	389.7

Some observations:

- I have counted only the lines of the files that I have written (for those that might say that I've counted the lines of the open source frameworks ;)
- Yes, those figures include the source code comments. I have yet to find a tool that only counts "logical" SLOC; in other words, these numbers reflect "physical" SLOC
- Very little code has been generated by third-party tools in this application (something that modern IDEs do all the time); the good news is that the Classic ASP Framework allows incredible productivity because of the high quality of the components and the ease of use... not to mention the fact that the API looks really similar to .NET... and that changes everything!
- The productivity jump from September 12th has to do with a particularly difficult point of the application that had to do with a graphical representation on the screen... once that was solved, the rest of the application "built itself" around that solution.

OK, hope that someone finds this information interesting... I do! It was real fun to do this application. I would love to show you more about it but, you know, this is proprietary stuff and NDAs apply here :)