Robert L. Bogue
MS MVP, MCSE, MCSA:Security
317-844-5310
Rob.Bogue@ThorProjects.com

The Value of Imperfect SharePoint Solutions

When we in information technology endeavor to create a solution we've been taught to produce complete solutions. We're conditioned to ignore solutions that don't completely solve the problem. We strive for that perfect round peg to put through the perfect round hole.

But reality is far from this pristine world. In the world of IT there are rarely round holes -- they're mostly round with a bit of a corner on one side. And the pegs aren't exactly cylinders, they're more like cones with a few deformities.

With some platforms, such as SharePoint, where we have the ability to quickly create solutions that mostly fit the problem -- and create a lot of value -- we should do that, create the solution that works and move on.

In this article we'll talk about some of the imperfect solutions that you can create with SharePoint -- and why they're the right thing to do.

When Partial Solutions Work

Before we get into good examples of imperfect -- but valuable -- SharePoint solutions it's important to deal with that nagging feeling of doom that may overcome you as we begin to talk about solutions that don't perfectly solve the problem. You've seen these solutions before and they have caused pain.

We've all seen or heard stories about Excel spreadsheets that have become unwieldy monsters that no one -- not even the creator -- really understands any more. We've seen the polka dot pattern of departments using Access databases to store information. Perhaps we've even seen custom-developed applications written in Visual Basic that don't integrate to any other system.

While recalling the horror stories of the applications that had to be rebuilt or which grew well beyond their original intent, it's easy to get lost in the effort to recover these situations. It's easy to forget the reason that these solutions were created in the first place: because they met a valuable need of the organization.

When creating imperfect solutions with SharePoint it's often important to realize that there is a problem to be solved -- and one that will be solved -- whether SharePoint is involved or not. Unless you've taken away Excel, Word, and Access from your customers, it's likely that they'll find an even less perfect way to solve the problem with Excel or Access. So when considering the solutions, consider the alternatives.

Recycle Bin

The subject of a trash bin, or more politically correct recycle bin, is one which has plagued SharePoint, and one that will be fixed in the next version (WSS V3). However, it's a good example of imperfect solutions to the problem. There are a few good recycle bin implementations for SharePoint available -- many for free and a few that are not.

One approach to the recycle bin, which is outlined in the MSDN article, "Add a Recycle Bin to Windows SharePoint Services for Easy Document Recovery," uses a mirrored document library, the SharePoint events model, and some changes to the JavaScript files which drive SharePoint. The solution does a good, but not perfect, job of being a recycle bin with the limitation that it requires double the storage since every file is stored twice -- once in the real document library and once in a shadow document library. This isn't a perfect solution to the problem; obviously it would be better if the storage needed wasn't duplicated and it requires a fair amount of technical expertise to get installed -- however, it's a workable solution.

Joel Olson and his colleagues within Microsoft's internal IT department faced the challenge by creating The "SharePoint Recycle Bin" which is available on GotDotNet.com. This approach leverages an ISAPI filter, the same technology which is at the heart of WSS V2. ISAPI filters are great technology but are not for the faint of heart. When a file is deleted, it's actually moved to the file system making it easier to restore than from a full backup but certainly not something the end user could do. Couple that with the registry entries and other miscellaneous configuration that isn't straightforward and it becomes pretty easy to see how this solution too has its challenges.

A third approach provided by Todd Bleeker of MindSharp is a List Template based solution, which uses some fancy JavaScript tricks to rename and hide deleted documents rather than deleting them. With all of the trimmings Bleeker's solution is easy to install but it has the limitation that it works for the Web only -- use WebDAV web folders and it won't work. So Bleeker's solution too is an imperfect solution to the problem of recycle bins.

None of these solutions is perfect, however, the solutions are workable. They all solve the core problem of users accidentally (or purposefully) deleting files. It's just that none of them covers it perfectly.

As a consumer you have the option of purchasing a commercial solution, like AvePoint's TrashBin. But there's a non-trivial cost to this option even though it may have fully resolve all of the issues that have been identified with the above solutions.

In the world of SharePoint, it bears considering that you may not need a perfect solution, just a workable solution.

Workflow

In the versions of Microsoft Windows before Windows 95, multitasking was supported but supported in a cooperative way. For multitasking to work, each application was responsible for giving up control when it could so other applications could run. This worked well when every application yielded control frequently, but even one application that didn't behave well made the whole system seem sluggish.

The workflow solutions that are supported with WSS V2 are largely collaborative workflows where documents are voluntarily moved from one location to another -- or meta data is changed from one value to another. As solutions go this is pretty powerful. If you're willing to work with open security and allow users to see everything it's possible to create processes based on little more than document meta data and some shortcuts.

CorasWorks is a good example of a vendor that has leveraged this technique to create workflow solutions in SharePoint. Their showroom has a series of solutions that highlight the approach of cooperative workflow. For instance, the Vacation Approval solution shows one workflow where vacation requests are approved. Nothing prevents an employee from submitting their request in an approved status -- other than perhaps the threat of a reprimand.

However, this cooperative approach doesn't work well when there's sensitive information that must be maintained in secret or when it's not acceptable to have to manually reset meta data on the document as approval is granted. In these cases, the collaborative (or manual) workflow process will break down.

There are commercial workflow solutions for SharePoint. Two of the most popular are K2's K2.Net and Captaris' Captaris Workflow, formerly TeamPlate. These tools allow for the creation of less cooperative and more prescriptive workflow solutions. However, neither product is without a learning curve, nor are they inexpensive.

There are many situations that require a process-based and non-cooperative workflow model, but there are many more where the potential risk for someone incorrectly submitting information, or the loss if they do is so low that it doesn't require a formal structured workflow -- and all of the costs associated with that infrastructure.

The next version of SharePoint (WSS V3), will include substantially better formal workflow built in -- and better support for third-party vendors -- which will ultimately allow for structured workflows with a lower cost. So the question is can you live with cooperative workflow solutions now and skip the expense of a formal workflow tool?

Putting it Together

Putting together a solution in SharePoint sometimes means compromises. It sometimes means forgoing a large upfront cost either to wait on a product enhancement or to wait on a compelling need that forces the investment to be made. Spending thousands of dollars on a recycle bin solution that may not be necessary or tens of thousands of dollars on a workflow solution may be the answer for the problem you're trying to solve. On the other hand, perhaps the difference between accepting the limitations of the basic solution and the cost of the more complete solutions are too great.

Take the agile software development approach by looking for solutions that solve a tangible part of the problem, learn more about the problem, and then try to solve more of the problem. Repeat this until the solution is good enough that the cost of refining it exceeds the amount of value to be gained by doing the refinement.

If you can do this you'll be less frustrated with SharePoint's limitations, you'll solve more problems, and have happier users because of it.

About the Author

Robert Bogue, MCSE (NT4/W2K), MCSA:Security, A+, Network+, Server+, I-Net+, IT Project+, E-Biz+, CDIA+ is the president of Thor Projects LLC. He has contributed to more than 100 book projects and numerous other publishing projects. He was recently honored to become a Microsoft MVP for Microsoft Office SharePoint Server. Before that Robert was a Microsoft Commerce Server MVP and before that Microsoft Windows Servers-Networking MVP. Robert blogs at http://www.thorprojects.com/blog You can reach Robert at Rob.Bogue@thorprojects.com.