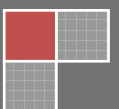


2009

SharePoint Workflow Options

and what to consider when choosing
between them.



Introduction

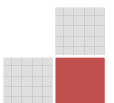
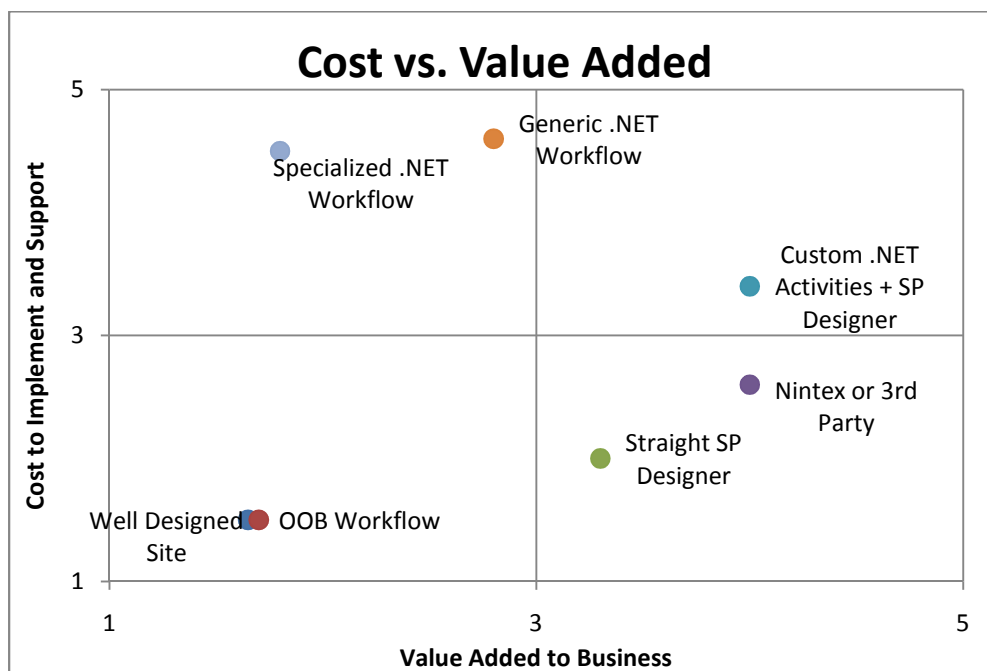
When it comes to building SharePoint workflow into a site, options are plentiful. Choosing the right option can make a world of difference to your project's bottom line and your client's satisfaction. In this paper I'll go over four "no-code" solutions and three .NET coding options and try to give you some consideration points to help you choose between them.

Who Should Read this Paper

- Governance committees when making decisions about what methods of workflow to endorse and support
- SharePoint Site Owners
- SharePoint .NET Developers
- IT Managers wondering what SharePoint Workflow can do for them
- My wife, my family members, & anyone else wondering what it is I do for a living

The Options

The options in this section are presented in the order in which I recommend them. That is to say, in general I recommend using the first option from this list that satisfies your needs. As you read through them, hanging on every word, keep in mind that these options are not mutually exclusive. For example, option 1 is good site design and option 3 is SharePoint Designer workflow. If you find yourself putting SPD workflows in a poorly designed site, that's bad. Your customers are not likely to be any happier at the end of the project than they are at the beginning.



Option 1: Good Site Design - Hey, That's Not Even Workflow.

Remember, the easiest workflow to maintain is one that you avoid building. It's not uncommon for a client to specifically ask for workflow when in fact what they want can be accomplished with a well designed site and some properly configured alerts.

Consider:

- Both a task list and an issues list can be configured to send email when the item is assigned. Your client may not need a "task" list, but keep your ears open for requirements that can be met by customizing these lists.

"A user fills out this request form. The supervisor gets notified with an email that a new request was submitted. Then the supervisor checks the employees' workloads and assigns it where appropriate. The supervisor emails the form to the assigned employee."

Start with a task list, add & rename fields to create the request form. Set up an alert for the supervisor for any new entries. Enable email notifications on the list. Top it off with some custom views: one for the employees that filters for active tasks assigned to them and one for the supervisor that groups the active tasks by assigned to.

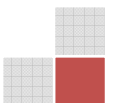
Need a work log to go with the request? Use the issues list instead of the task list and make the "comment" field required.

- Alerts can be set up on views. If the client simply wants notifications when items hit specific states, this may be adequate.
- The client may be asking to run a workflow on a form (Word, InfoPath, Excel) and the flow may depend on data within the form. Consider if it may be possible to convert the form over to a SharePoint list. It's easier to get at the data if it's in a native list and not inside a wrapper.

NOTE: You may also be able to get at this information if it is an InfoPath form.

- When a user asks for a notification, they may not necessarily mean getting an email. Emails can be hard to keep track of. Perhaps a web part on a site's home page is a better solution. You can accomplish these on-the-site notifications with list view web parts, or content query web parts. If you're using this option, consider setting the target audience for the web part so it's only visible to the users that care about it.

Example: Use a content query web part to show all the pending items in a site collection on the home page. Set this web part's target audience to the various approvers groups in that collection. Result: only approvers in the site collection will see the web part and they will only see items which they have the authority to approve.



- You can turn on “requires approval” without actually using the built in approval workflow. This can be especially handy for users that want an approval check before an item goes live but don’t want to deal with emails and task lists. Just turn on approval and create an on-the-site notification for the approvers group.

Option 2: SharePoint Out of the Box (OOB) Workflow

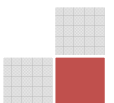
SharePoint comes out of the box with a few workflows: approval, three state, disposition approval, and collect feedback. These workflows are written to be generic and thus assist in a wide range of situations. Some configuration is possible as well, for example, the approval workflow tasks may be assigned to multiple users simultaneously or one at a time (parallel or serial approval). Overall, however, these OOB options are fairly rigid. They do well for those activities in your site that happen to work in the cookie cutter fashion, but beyond those situations, even seemingly simple processes can be outside the scope of this option.

Consider:

- The OOB approval workflow works well for the page and article content within a publishing site. When used in this situation, the approval workflow works in conjunction with versioning, the “requires approval” setting, and the publishing toolbar to provide a smooth user interaction for content publishing and approval. The approve/reject buttons are handled nicely right in the page and the approval process is kicked off appropriately.
- The OOB approval workflow works adequately in document libraries and image libraries, though it may be more overhead than your users want. Clients may instead prefer to turn on versioning and requires approval for the images and documents in the site and rely on the page’s “check for unpublished items” report to make sure content is current and live. In general this works well because that other content is accessed through the site’s pages.
- The Option 1 alternative exists for publishing sites too - Even in the pages library of a publishing site, clients may prefer to just use the versioning and requires approval option without the approval workflow being activated. This will give the users the save as draft, publish, approve, and reject buttons where appropriate but will not require workflows to be started or emails to be sent.
- The Office 2007 product suite has workflow support built in to Outlook, Word, and Excel.
- These OOB workflows may not be useful to users except where used in document libraries where only MS Office documents are used and the client’s all have Office 2007.
- The OOB workflows have no support for conditional paths or branching logic.
- The OOB workflows do not interact with other lists (except the task list).
- There is no way to customize the email text being sent by the OOB workflows.

Option 3: SharePoint Designer (SPD) Based workflow

Using the [now free SharePoint Designer](#), even a novice user can begin to create truly customized workflows. It’s easy to string a few activities together and create a basic workflow that fills a gap that Options 1 and 2 leave open. A SPD expert can create fairly advanced applications using a combination of these first 3 options.



Consider:

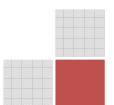
- SPD allows you to build branching logic into your workflow.
- SPD allows workflows to interact with other lists – lookup values, create/update/delete items.
- SPD does allow you to customize your email text, though there are some limitations in particular around hyperlinks.
- OOB SPD workflows may only interact with the site in which they are executing. There is no cross-site or external interaction allowed. You may get around this limitation by custom coding your own activities in .NET (read on).
- There is no moving a SPD workflow. It's built for a list and it runs only on that list. The only things here that are reusable are the techniques your SPD developers learn. If you have a repeatable process and you want to reuse the workflow, you'll need to duplicate it.
- Now let me backtrack on the previous point... You can encapsulate a fully functioning workflow in a site template. If what you've built is a reusable application –ex. a site that encapsulates your project methodology, or a site that helps manage major crisis events – you can save that site as a template with the workflow intact. Just check the include site content box when you are creating the template. This works because SPD workflows are just content within a workflow library. You can then reuse that site template, even export it and use it in different site collections or on a different server.
- Remember SPD can be a dangerous tool and most governance plans will keep it out of the hands of the general population. Ambitious clients might put this tool in the hands of a few trained site owners, but most will keep it under control of their IT departments.
- You can design your way into an infinite loop if your SPD workflow is configured to run when an item is updated and it is the job of that workflow to update the current item. It won't crash your server, but it's going to slow things way down until you fix it.

Option 4: Third Party Workflow Plug-ins

There are several vendors in this space and, before choosing one, you should complete the due diligence required to settle your mind and the minds of those who control the purse strings. However, for the purpose of this paper, I'm going to focus on the one that I like best: Nintex (www.nintex.com).

With a third party tool, there are some common traits that most clients are hoping to gain. Below, are what I think are the top 5 things a client will be looking for in a third party workflow add on and how I think Nintex addresses these desires.

Common traits that clients are looking for in a third party solution.	How Nintex addresses the need.
The ability to separate their workflow tool from SPD. Clients may not want their general site owners using SPD, but there is often still a business need for these users to create custom workflows within their sites.	The Nintex workflow designer is browser based, so once the server piece is installed, the designer will be available to all your users without any client side install or licensing requirements. It duplicates all, or nearly all of the SPD



	activities that are available and adds in several more key activities for interacting with SharePoint such as creating & deleting sites, sending documents to a library in another site, creating new lists, kicking off other workflows, setting item level permissions, and the ever sought after looping structures.
Simpler workflow creation. Even if users have access to SPD, they may feel that the workflow design interface there is not intuitive enough.	The Nintex designer has a nice flowchart like layout where users can visually see the branching logic and interact with the design.
External systems integration and interaction. Your client may just need to do something simple, like copy a file to a file share, or they may want full blown ERP systems integration.	Nintex may not get you all the way to ERP systems integration without some custom coding, but it supports a few things that can help put you most of the way there. <ul style="list-style-type: none"> • Support for interacting with Active Directory and Exchange. • You can dump files to a file share. • It also has the ability to call a web service and take actions based on the response. While getting the syntax of the web service call and handling the response may not be trivial, this extensibility point is present and very powerful. Couple this with SharePoint's native lists web service and your external systems can also call back to trigger events in SharePoint.
Reusable workflows. Clients that identify a common procedure, and take the time to implement it in a workflow would like an easy way to reuse that workflow throughout their environment.	Nintex allows you to save a workflow as a template, which then makes it available again and again.
Portable workflows. Clients that utilize dev/staging and production environments may wish to test their workflow out in a preproduction environment before going live with it in production.	Nintex workflows can be exported and imported on any SharePoint environment where Nintex is installed. You may have varying levels of reconfiguration to do after importing a workflow if it relies on list names or group names, but for the most part, this reconfiguration is straight forward.

Here's some extras that I think make Nintex an excellent value.

- "Lazy Approval" – Perhaps one of the most powerful features that Nintex has given their customers. When an approval activity sends out its email notification to the approvers, Lazy Approval allows those users to simply



reply back to the email using a predetermined text phrase. Site owners maintain this list of accept/reject phrases and can add to the list as desired. It's a beautifully smooth user interaction that works asynchronously and disconnected. Almost every company out there has 24 hour anywhere access to their email systems, but an internal SharePoint instance may only be available while logged in, either inside the firewall or over VPN. Lazy Approval can be done over your BlackBerry.

- A nice rich text email editor – for all those custom notifications you want to send out. If you've created email using SPD, then you should understand the power of this feature. It works well and it looks good. Don't care for the look? You can custom brand the headers and footers – brilliant. You can also save your creations as email templates to reuse.
- Extensibility – need some other activity or some kind of specialized external interaction? The Nintex API is for you.
- Copy and Paste within a workflow – I found myself putting together a 4 tier approval workflow with each tier behaving the same way. I rebuilt the interaction twice before realizing I could copy and paste the whole section and then just plug in the changes to the email notifications and the approvers groups.
- Better workflow history – Getting audit type information (who did what when) out of the OOB or SPD workflows is possible, but it's not very intuitive. Nintex gives you this information in a nice layout along a very intuitive path.

Option 5: Custom Coded .NET Workflow (.NET WF) Solutions AKA. The Last Resort

Options 1-4 can all be referred to as No-Code solutions. It doesn't mean they're trivial, it doesn't mean you won't need IT involvement to create them, and it doesn't mean that you won't need IT to help support them. It simply means there is nothing there that gets compiled down to an executable or a .dll that needs installed on the server. However, when you are using Options 1-4, you are building your applications on a very solid foundation and on a very rapid application development platform. Your IT department's involvement is drastically reduced when you use options 1-4 as these applications will take fewer developers and require a much smaller QA effort. A single, strong SharePoint developer, using options 1-4, can accomplish in a day or two what may take a small team of .NET developers several weeks to scratch code.

Now that the dire warnings have been sounded, let me say this – there is a time and a place for SharePoint .NET workflows development. This platform is wide open though, so how do you make sure you are directing your development efforts to where the best value is returned?

As I see it there are 3 flavors of .NET WF development. Your .NET development dollars should be spent in the first two flavors when the ROI makes sense and on the last flavor only while under duress.

Flavor 1: Custom Coded Activities.

Whether you're using SPD or a Nintex type workflow designer, you'll have the ability to write custom activities to plug into your system. The key here is to just write the missing activities that you need. You're developers



SharePoint Workflow Options

will be writing many fewer lines of code that will in turn require a much smaller and more focused QA effort. The ROI for such an effort is much more achievable.

Additionally, because your developers are coding up small, reusable activities instead of big intensive applications, your clients will end up with a much more flexible solution that's more easily maintained.

The user community has already made several SPD activities available for free on CodePlex (<http://spdactivities.codeplex.com/>). If you are using the SPD option for workflow, I highly recommend you enhance your system by installing these activities.

Nintex (and other third party options with APIs are likely to as well) has an active developer forum where you may get in touch with other developers to seek advice and get examples. You may find some free additions these products out there as well.

Consider if:

- You can meet most of your needs with Options 3 or 4, but there is a hole in your coverage that keeps resurfacing.
- You are working on a project and can meet 90% of the workflow requirements with native SPD or Nintex, but you just can't quite get all the way there natively. In this case, use the 90% native and concentrate your coding efforts on only the 10% that's left. That 10% may be highly specialized, but you've saved an enormous amount of effort on the rest of the project.

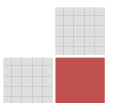
Flavor 2: Reusable Custom Coded Workflows

This type of workflow is installed on the server and is made available throughout your environment. Your users are able to initialize and configure these workflows within the list or library settings, just as they are able to configure the OOB workflows.

These workflows are more complicated to create and more intensive to test than the first flavor. However, if you locate the right project, your ROI may still be reasonable. The MOSS SDK comes with some excellent samples & starters for these. One in particular that your clients may find useful is the multistage approval workflow. It's the perfect example of a highly reusable workflow. It's also a great example of just how complicated these workflows can be to code. Once the workflow is installed, the effort you've spent can see value returned from across your entire SharePoint implementation.

Another highly reusable workflow example would be a modified disposition workflow that archives your document to a third party document repository that your organization already has in place.

Consider if:



- You've identified a self contained yet highly reusable workflow for your organization, such as the multistage workflow, or archiving to a third party document repository.

Flavor 3: Specialized Custom Coded Workflows

This is the most expensive flavor of workflow to implement. Under most circumstances the ROI cannot be calculated. The idea here is that you are custom coding a workflow that is not reusable. It may be a one off created specifically for a particular form or list. If there are many steps or variables, you can easily rack up several thousand lines of code and QA can quickly become a nightmare.

There may be reasons where you would want to use this flavor. Maybe the ROI is offset by a legal obligation. Maybe you are packaging up a solution for resale and don't want any licensing overhead. Maybe you're getting paid by the hour and you enjoy self inflicted pain.

If you can satisfy your client through any other option or flavor, do it. They'll be much happier with you in the long run.

Consider if:

- ROI is not the deciding factor. External factors such as legal obligations come into play.
- You intend to resell your solution.
- You're crazy.

Conclusion

SharePoint workflow is an extremely powerful tool if done properly. Choosing the correct workflow option can economically add a tremendous amount of value to your client's SharePoint environment and lead to very happy customers. However, if you choose the wrong development option, you can end up with a solution that doesn't meet your client's needs, blows your project budget, overruns your project deadline, or all three of these. The trick to picking the most effective workflow solution is to choose a workflow option that meets your client's needs and requires as little scratch development as possible. The above sections should assist you in evaluating your options, but in general keep these points in mind during the process:

- Always begin with a good foundation in the form of a well designed site.
- Use item approval and the OOB workflows as they meet your client's needs.
- If you need additional workflow functionality, use SP Designer or purchase a third party add in, such as Nintex.
- If you can't meet all your client's needs using SP Designer or another off the shelf product, come as close as you can and then bridge the remaining gap using .NET to write your own Designer activity or third party plug in.
- Build full fledged .NET workflows only when their reusability justifies the effort or as a last resort.

