**Why to avoid full-trust-solutions in SharePoint online**

In SharePoint 2007 Microsoft introduced the ability to extend SharePoint’s capabilities with **‘Full Trust’** or **‘Farm Solutions’**, which was a major improvement in standardizing the packaging and deployment of customizations.

Such a solution could be sets of custom web parts, timer jobs and web templates written in server-side languages like C#. All code had ‘Full Trust’ (or at least more than required) permissions on the SharePoint farm, therefore the name ‘Full Trust Farm Solution’ was adopted.

Microsoft introduced the concept as it wanted a standard way that companies and partners could customize SharePoint and add new features. SharePoint was starting to become very popular around the time of the 2007 release, and as a result many people were looking at how to extend or expand what came ‘out of the box’. Full Trust Farm Solutions were the perfect answer. Or so it was thought at the time.

This approach gave developers the tools to customize SharePoint in any way they needed, but it did have several drawbacks. As all code runs on SharePoint servers, poorly written code could decrease performance significantly. These performance issues would affect every application running on that server, not just SharePoint. Full Trust Farm Solutions also have security concerns, because the code has full control within the SharePoint farm. Badly architected code can thus in theory affect the entire SharePoint solution.

**A victim of success**

Microsoft never anticipated that Farm Solutions would be such a big success, but they were and became immensely popular with developers.

To try and address some of their drawbacks Microsoft introduced ‘Sandbox Solutions’ in SharePoint 2010. This approach offered a subset of the functionality of Full Trust Farm Solutions, with code being forced to run in its own separate sandboxed process.

However, the functional limitations meant Sandboxed Solutions were always seen as a poor relation to Full Trust Farm Solutions. Microsoft decided to deprecate/discourage Sandbox Solutions in SharePoint 2013, meaning that they should no longer be used for user code.

**A new approach with SharePoint 2013**

Microsoft introduced the ‘App Model’ in SharePoint 2013, radically changing the whole approach to custom code development. All customizations are either run in the browser using JavaScript, or on another platform with externally hosted Apps. No server-side code is run on the SharePoint servers anymore, meaning badly written code cannot affect core SharePoint services.

Apps can be deployed to both SharePoint 2013 ‘On Premises’ environments as well Office 365. In Office 365 the App Model is the only approach developers have, as Microsoft can be much stricter about the code it allows to run on its own servers.

**FIVE REASONS TO AVOID ‘FULL TRUST’ COMPLETELY**

Let us look at five key reasons that ‘Full Trust’ solutions should be avoided, and why the alternative App Model is much better:

**1. Memory issues**

Server-side code that talks directly to SharePoint is prone to memory leaks. If SharePoint ‘best practice’ code guidelines are not followed strictly, the memory usage of the IIS web sites hosting SharePoint applications will increase, decreasing performance of the SharePoint farm.

SharePoint Apps don’t have these issues, because all code is run outside of the SharePoint farm.

**2. Security**

Full Trust Farm Solutions run either in the context of the web application (IIS Application pool) or the timer service (Farm service account). In either the code has almost full access to the SharePoint farm, and even to other web applications. So, for every Farm Solution that is deployed, security checks need to be performed manually to make sure no malicious code is being run.

SharePoint Apps run in their own context, the so-called ‘App Web’ or a ‘Provider Hosted Web’. Permissions to existing webs need to be granted explicitly.

**3. Not Cloud proof**

Farm Solutions cannot be deployed to Office 365/SharePoint Online. The only way to leverage Farm Solutions in the Cloud, is by creating a ‘Provider Hosted App’ and point to an On Premises solution.

A SharePoint App can be deployed to both On Premises as well as to Office 365.

**4. Harder to upgrade**

Farm Solutions are tightly coupled to SharePoint and the code runs in the same context as SharePoint, making them harder to upgrade. The more custom code that exists in a Farm Solution, the more effort is required to upgrade. This often means that the entire SharePoint farm is unavailable during upgrade of the customizations.

SharePoint Apps on the other hand are loosely coupled with SharePoint. Apps have a minimal dependency on SharePoint, making them easier to upgrade.

**5. Dependent on .NET framework**

SharePoint is built on the .NET framework. Full Trust Farm Solutions are required to be written in the same .NET version as the version the solution is deployed to.

For example, SharePoint 2010 is built on .NET 3.5, meaning all solutions must use the same version. They cannot take advantage of the improvements in versions beyond .NET 3.5.

SharePoint Apps, in particular ‘Provider Hosted Apps’, can be written in any language and hosted on any platform. They can even be hosted on a Linux server and written in PHP.

**APPS ARE THE FUTURE**

Full Trust Farm Solutions allow almost any type of functionality or customization to be created, however they have big drawbacks with security, performance, and upgradeability. Therefore, Microsoft is heavily pushing Apps as the best practice solutions to create custom functionality for SharePoint. Not only are they a better way of creating great solutions, but they can be deployed easily to both On-Premises and Office 365 environments.

# Transform sandbox solutions to the SharePoint Add-in model

Transforming your sandbox solutions to the SharePoint Add-in model involves analyzing your existing extensions, designing and developing your new SharePoint Add-ins, and then testing and deploying your add-in in your production environment.

**Note**

[**Code-based sandbox solutions were deprecated**](https://blogs.msdn.microsoft.com/sharepointdev/2014/01/14/deprecation-of-custom-code-in-sandboxed-solutions/) in 2014, and SharePoint Online has started the process to completely remove this capability. Code-based sandbox solutions are also deprecated in SharePoint 2013 and in SharePoint 2016.

## Code-based sandbox solutions in SharePoint

Sandbox solutions are customization packages that can be used to deploy customizations to SharePoint on the site collection level. If a sandbox solution contains code, it has been executed in a special isolated process with a limited set of APIs to access SharePoint services and content.

There are two types of Sandbox solutions:

* Code-based sandbox solutions, which contain a custom assembly in the package.
* Declarative sandbox solutions, which only contain XML-based configurations and related assets.

Declarative (XML-based) sandbox solutions can be further divided into the following types based on their use cases:

* Site template – Generated by using the “Save site as a template” functionality from existing sites.
* Design package – Generated by using Design Manager from the publishing site.
* Custom sandbox solutions - Created in Visual Studio, for example, for branding assets, and do not contain assemblies.

Code-based sandbox solutions can be further divided into the following types based on their use cases:

* Declarative sandbox solution with empty assembly
* Sandbox solution containing InfoPath form with code
* Code-based sandbox solutions with customizations such as web parts, event receivers, and/or feature receivers
* Sandbox solutions with custom workflow action

When you plan to move away from sandbox solutions, you should evaluate the functional and business requirements of a specific solution and determine the future design direction based on that.

## Planning the transformation process

When you transform your sandbox solutions to the SharePoint Add-in model, you want to ensure that the impact on your users is minimal. Carefully analyze your current sandbox solutions, and then design your new SharePoint Add-in to meet the needs of your organization. We recommend the following process to ensure a successful transformation.

### Readiness

Learn about:

* The SharePoint Add-in model, different kinds of add-ins, and hosting options. For more information, see [SharePoint Add-ins](https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/sharepoint-add-ins) and [Office 365 development and SharePoint PnP solution guidance](https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/office-365-development-patterns-and-practices-solution-guidance).

### Solution assessment

Analyze the functional and business requirements by:

* Identifying deployed sandbox solutions in your current environment for which you can use either of the following tools provided as open source by the SharePoint PnP team:
	+ The [SharePoint Sandbox Solution scanner](https://github.com/SharePoint/PnP-Tools/tree/master/Solutions/SharePoint.SandBoxTool) or [video](https://www.youtube.com/watch?v=pK4p2mGYXpU): A tool offering various options and a detailed inventory.
	+ A [specific sandbox solution inventory script](https://github.com/SharePoint/PnP-Tools/tree/master/Scripts/SharePoint.Sandbox.ListSolutionsFromTenant): A PowerShell script that gives you a basic inventory.
* Reviewing requirements with your users. Consider asking your users to demonstrate how they use the existing sandbox solutions to perform their daily work.
* Identifying, documenting, and designing new functionality to include in the new SharePoint Add-in. Consider reviewing your list of new feature requests from your users for additional ideas.
* Identifying unused features, and agreeing with your users to omit this functionality from the new SharePoint Add-in.
* For each solution, determining whether to replace it with a SharePoint Add-in or implement it either by using out-of-the-box capabilities or an alternative solution.

### Solution planning

Design the new application by using the SharePoint Add-in model based on:

* The requirements gathered in the **Solution assessment** step.
* Your analysis of the existing code. During your code analysis, consider identifying portions of the code that can be dropped (for example, the code is no longer being used, or the requirements have changed).

### Develop and test the SharePoint Add-in model version of your application

This is usually the most time-consuming step in the transformation process.

### Deploy your new add-in

Ensure that your deployment is stable, and send appropriate communication to your users.

## Replacing sandbox solution customizations

Following are typical customizations that are included in the sandbox solutions and potential transformation options. We are looking at adding more information for each of the customization types so you can have real-world examples about the transformation options.

| **Customization** | **Transformation options** |
| --- | --- |
| Declarative solution with empty assembly | You can control assembly creation from Visual Studio solution project properties. For more information, see [Remove the assembly reference from your Sandbox solution created in Visual Studio](https://support.microsoft.com/en-us/help/3183084/remove-the-assembly-reference-from-your-sandbox-solution-created-in-vi).**Important:** When you use the SharePoint Sandbox Solution scanner, the scan output lists which solutions have an empty assembly, and the tool creates updated sandbox solution packages for you in which the assembly is dropped. You can then simply replace the existing sandbox solution with the updated one. |
| InfoPath form with code | If you have published an InfoPath form from the InfoPath client that contains code, it’s actually published to SharePoint as a sandbox solution. This means that the form code is actually executed by the sandbox engine in SharePoint.Moving away from the code-based InfoPath forms is dependent on the actual business use case. There are multiple options from generating custom UI as add-ins or utilizing other form techniques.For more information, see [Fix InfoPath in sandbox solutions](https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/sandbox-solution-transformation-guidance-infopath). |
| Web part | Web parts are typically converted either to add-in parts or they are implemented with fully client-side technologies by using the JavaScript embed pattern.For more information, see: [Customize your SharePoint site UI by using JavaScript](https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/customize-your-sharepoint-site-ui-by-using-javascript) [Create add-in parts to install with your SharePoint Add-in](https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/create-add-in-parts-to-install-with-your-sharepoint-add-in) [How to update your SharePoint pages via the embedding of JavaScript](https://channel9.msdn.com/blogs/OfficeDevPnP/JavaScript-embedding-demo) [Cross site collection navigation](https://channel9.msdn.com/blogs/OfficeDevPnP/Cross-site-collection-navigation) |
| Visual web part | Visual web parts are transformed in similar ways as normal web parts. User controls used in visual web parts must be replaced because in sandbox solution cases, it's included inside of the assembly. |
| Event receiver | Event receivers can in many cases be replaced with the remote event receiver implementation. Remote event receivers do, however, need to hosted on some platform, typically on specific provider-hosted add-ins.For more information, see: [Use remote event receivers in SharePoint](https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/use-remote-event-receivers-in-sharepoint) [How to use remote event receivers for your SharePoint add-ins](https://channel9.msdn.com/blogs/OfficeDevPnP/How-to-use-remote-event-receivers-for-your-SharePoint-add-ins) |
| Feature receiver | Feature receivers are typically replaced with a remote API based operation, such as using CSOM or REST for applying the needed customization or configuration at the site level. If a needed API is missing from the remote APIs (CSOM/REST), report this gap by using [SharePoint UserVoice](https://sharepoint.uservoice.com/).Feature receivers are used, for example, to set a custom master page or theme to a site when they are activated. These kinds of operations can be easily replaced with remote code-based solutions or by using [PnP PowerShell](https://github.com/SharePoint/PnP-PowerShell/blob/master/README.md), which provides easy commands for controlling site configuration. |
| Custom workflow action | The typical code migration path for these kinds of customizations is to use SharePoint workflows or alternative solutions such as Microsoft Flow or third-party solutions. |

## Removing sandbox code from your site

When you deactivate your existing sandbox solution from sites, any assets or files deployed by using declarative options will not be removed. If you have used sandbox solutions to introduce new code-based web parts, those functionalities will be disabled from the sites. This means that the pages are still rendering normally, so there's no direct end user impact when the solution is deactivated, except removal of the code-based functionalities such as web parts.

## Removing support for code-based sandbox solutions

Support of code-based sandbox solutions will be removed from SharePoint Online by disabling code-based operations that execute from sandbox-solution-based code. This means that your sandbox solutions are not explicitly deactivated from the solution store, but any code-based operation will no longer be performed. Sandbox solutions will remain in activated status in the solution gallery. Features deployed by using sandbox solutions will not get deactivated automatically, which means that possible code associated to feature deactivation or uninstall handlers won't be run.

All declarative definitions in the sandbox solution will continue working after this change is applied in SharePoint Online.

**Convertible Objects:**

The following table describes the items will be converted via Metalogix tooling:

|  |  |  |
| --- | --- | --- |
| Sites  | Lists  | Items  |
| Permissions and Sharing   | Permissions and Sharing  | Permissions and Sharing   |
| Tile  | Documents  | Versions (5)xx  |
| Logo  | Items  |   |
| Pages  | Views  |   |
| Web Parts  | Content Types  |   |
| Managed Metadata Terms  | Managed Metadata Columns  |   |
| Declarative Workflows   | Declarative Workflows  |   |
| OOB Themes   |   |   |
| Site Columns  |   |   |
| Navigation  |     |   |
| Content Types  |   |   |
| Property Bag  |   |   |
| Quota  |   |   |

**Non-Convertible Objects:**

The following table describes the objects will not be converted the Metalogix tools and must be remediated by conversion team and site owner:

|  |  |  |
| --- | --- | --- |
| Sites  | Lists  | Items  |
| Running workflows or history  | Running workflows or history  | SSRS reports  |
| Custom user profiles or values   | Custom list forms – add, edit, view  | Draft version items are hidden with preceding \_   |
| Web Templates  | List templates  | Checked out / Locked with no previous checked in version   |
| Hidden Lists  | Hidden Lists  |   |
| Checked Out Versions - Pages  | Checked Out Versions – List-Library items  |   |
| 1st version checked out – Pages  | 1st version checked out – List-Library items  |   |
| Anonymous sites/pages  | Recycle Bins (both)  |   |
| Un-ghosted/Customized pages   | Access Request List  |   |
| Custom Apps/ 3rd party components, web parts  | IRM Settings  |   |
|  Custom site branding, design packaging, definitions, themes  |   |   |
| Information Management policies   |   |   |
| Full trust code / Sandbox solutions  |   |   |
| Custom events receivers  |   |   |
| Custom Managed Paths  |   |   |
| Alternate address mapping  |   |   |
| Usage Reporting  |   |   |
| Site Settings  |   |   |
| Vanity URLs  |   |   |
| Custom Web Parts  |   |   |

# How to Build a SharePoint Migration Plan

Although this isn’t necessarily a bad solution, you should first make a roadmap and plan each phase of the migration to ensure you’ve thought of everything.

## **Before Building your SharePoint Migration Roadmap**

What is a roadmap? I could pull out a definition from a sophisticated source or we could try to define it together.

Basically, a roadmap is exactly as it sounds. We are identifying the milestones we want to achieve before arriving to our destination.

But before we begin the roadmap, we obviously need to know what our destination is.

### **Define a Destination**

Before we create our migration plan, we should know what to expect after the project is over.

Define on paper the criteria for a successful SharePoint migration. When do you expect the migration to be complete? What content should absolutely be moved? Should it be branded? Etc.

### **Use the RMR Strategy**

Before we can create our roadmap, we need to know what we’re going to migrate.

**R**emove, **M**igrate and **R**ebuild.

#### **Remove**

These are the sites you do not plan on moving. Note the word Remove- this means we are explicitly deleting them or leaving them on the old SharePoint, if they are isolated in their own database.

The idea is that these Sites will not be moved to the new SharePoint farm.

#### **Migrate**

These are the sites that will be migrated to SharePoint. How? That’s up to you! Choose the type of migration you think is best for you.

Both SharePoint 2013 and 2016 offer deferred Site Collection upgrade, where the Site Collection administrator can upgrade it himself after an upgrade preview.

At a granular level, you could upgrade the Site Collection to 2013 or 2016, then use a combination of export/import to move individual sites. Though this could prove challenging depending on your SharePoint.

Finally, you could use following options to migrate. Again, how you choose to migrate is up to you.

* [Metalogix(Now Quest)](https://www.quest.com/metalogix/)
* [ShareGate](https://sharegate.com/products/sharegate-desktop)
* [SharePoint Migration Tool](https://docs.microsoft.com/en-us/sharepointmigration/how-to-use-the-sharepoint-migration-tool)

#### **Rebuild**

Another option is to completely rebuild the site in the new version. This usually happens to heavily customized sites in SharePoint 2007 or SharePoint 2010.

With new Web Parts and Apps available in SharePoint, there are probably a few sites that either won’t work or just need to be rebuilt to take advantage of a new architecture and solutions.

## **Building your SharePoint Migration Roadmap**

Now that you know what the RMR strategy is and have planned your migration by establishing the criteria for a successful migration, you are ready to create the roadmap.

Essentially, you could create a beautiful roadmap using **XMind**, **Visio** or other similar tools. But the method I use is a little more hand on, I use SharePoint!

Wait what? No, I don’t draw anything, but what I do is create a SharePoint list to help me identify the sites affected by the migration.

My objective is to list all the sites affected and identify whether they will be Removed, Migrated or Rebuilt. I also need to know that the action is approved and when it will be done.

### Create the SharePoint list.



I created a list called Migration Roadmap with the columns above. Within it I will enter all my sites from SharePoint 2007 or 2010, or whichever environment I am migrating from.

If you don’t have an inventory already available, you can easily export it using PowerShell. You could probably find some [scripts on codeplex as well like these ones](http://sharepointpsscripts.codeplex.com/).

I use the following command:



Technically you could even take this CSV file and import it into the Visio wizard and it will draw a diagram for you.

The migration action column has three choices: Remove, Migrate and Rebuild. I also enabled Content Approval on the list, so that the business could approve whenever we put the migration action to Remove.

### Add the Content in the SharePoint List to Create your Roadmap.



Of course, the list would be a lot longer in a non-fictional organization.

This is now your SharePoint migration plan or roadmap. You can export or create a custom view to show this list however you wish, whether for a manager or other users. All the information needed is listed here.

Now you either Remove, Migrate or Rebuild your sites in SharePoint.

There is a reason that I wrote this article after I talked about “[not wrapping the old with the new](https://sharegate.com/blog/designing-sharepoint-2013-architecture).” It’s not always going to be Remove, Migrate or Rebuild.

As I mentioned in the previous article, there are many new features and Web Parts that can alter your architecture. Some will fit this in the Rebuild migration action, and that’s fine, the important thing is to adapt the strategy to your needs.

With the new search and content search Web Part you are bound to do things differently. What I am trying to say is, take the time needed to carefully analyze the existing architecture before moving everything as is.

## Planning your SharePoint Migration

There are certain details you must prepare before you execute the roadmap.

* Who oversees the Remove portion of the roadmap?
* Who oversees the Migrate portion?
* Who oversees the Rebuild portion?
* What is the workflow to approve each of those actions? And who will be approving them?
* Who will be meeting the site owners, will it be the same person in charge of the action?

I created a “points” system to assign a business priority to the sites to be moved. A combination of the level of interest that a team has towards SharePoint (did they follow training, have they shown interest in working with SharePoint) and the number of people affected. This helped me prioritize the migration.

I am sure you will think of other things to plan for based on your organization and migration project, because there are never two the same.

What I will never stress enough, is that you need a plan and a roadmap, otherwise you will be stuck in the never-ending migration cycle, where at some point, someone is going to get mad and say that’s it.

# Building a SharePoint Governance Plan Before Migrating

## What is a SharePoint Governance Plan?

A SharePoint Governance plan is a set of rules that help to facilitate the use, maintenance and operations of your SharePoint. The plan helps set expectations and guidance for your team, as well as the end users.

The problem I see in most Governance plans is that they try to do, or to say too much. The result, no one reads or follows it and you’ve wasted weeks writing it. Why? Because most land on the [TechNet section dedicated to SharePoint Governance](https://technet.microsoft.com/en-us/office/dn788776?query=sharepoint+governance).

I think the information Microsoft provided is amazing, however it has way too much content, which leads the person in charge of Governance to sometimes overdo it.

If a SharePoint Governance plan is 100 pages long, it simply will not drive anyone to read it or refer to it.

## Using Enterprise Wiki for Building a Governance Plan

If you already have SharePoint, a fun way to start Governance is by using the Enterprise Wiki. This way, you can easily link your pages.

Start your Governance plan by topic, and tag each one with a Wiki identifying the “topic” pages.

## Topics that Should Be Defined

There are certain things that absolutely need to be defined, but remember that it is not an inventory of your environment!

Typical example: The Governance plan should state what SharePoint environments will be used, with a short description as to what goes in it, or what it is used for.

|  |  |  |
| --- | --- | --- |
| **Type** | **Description** | **Owner** |
| **DEV** | Development environment limited to SharePoint developers. | Devs |
| **INTEGRATION** | This environment helps to test, to package and the integration of solutions. | Devs |
| **TEST** | Application tests and solution sign-offs from business users. | IT SP Team |
| **PRODUCTION** | Live environment with tested packages and solutions | IT SP Team |

As you can see in my example above, I am not listing URLs or the number of servers, etc. That is not the role of the Governance plan.

Here is a list of Governance topics I make sure to have before beginning a SharePoint migration:

|  |  |
| --- | --- |
| **Topic** | **Description** |
| **Site Request** | How does one ask for a new site and what goes into it. |
| **Site Template** | What Templates are available (including custom) and what are they used for. |
| **Site Management** | Definition of the allowed management of the site for the Site Owner. Is it free for all, slightly controlled or managed by an Information Architecture? |
| **Unused Sites** | What are they and what happens to them? |
| **User Agreement** | Whenever they get a new Site or Site Collection, Site Owners must agree to access the site. |
| **Security** | How is it managed? |
| **Social Policies** | Ratings, Tagging, Status etc… The My Site Profile privacy. |
| **Multilingual** | It's very important to explain what users should expect for multilingual needs in SharePoint. Variations, MUI or 3rd party tools? |
| **Support Model** | How is support defined? Who does the End User ask to get answers? |
| **Naming Convention** | Databases, Servers and Web Applications should be named using a designated convention. |
| **Backups and Restores** | What is the schedule for backups and when are restores tested? |
| **Development** | Define how thr development of new features will be done. |
| **SharePoint Designer** | Will it be used and if so, by who and how? |
| **Archiving** | Is there a plan for Records or will you just be eliminating old content over time? Link to a file plan if it exists. |
| **Communication Plan** | Constant communication with the End Users needs to be there to ensure the success of the project. |
| **Training** | Define what kind of training will be available for site owners, administrators and end users. |
| **Etc.** |   |

The key to successful SharePoint Governance is to keep it simple. Every topic I mentioned above should stay very high level to help the business know what to do by referring to this document.

## Keeping the Support Model Simple

Instead of getting into super complicated definitions of what the support model should be in your organization, use something like PowerPoint.



This way, I am sure everyone will understand the flow of support.

Don’t forget, if you chose to use a Wiki for your SharePoint Governance plan, it will be easily accessible from the Team Sites.

One effective way to do this is to add an option to contact support through the Site Actions menu.

## Creating a SharePoint Information Architecture

I know what you’re thinking- Information Architecture? Wasn’t this an article on SharePoint Governance? Bear with me here. Information Architecture is like the best friend of a Governance plan, they complete each other!

Think of it this way- if you give a document that is so well defined that the developer, or whoever is building the SharePoint environment, can take it and create the sites exactly how you needed them; then you have an Information Architecture.

Whenever I create an Information Architecture, I want to make sure that everything is covered. Content types, Site Columns, Sites, Document Libraries and everything else required.



Using a combination of Excel and Word, I can list everything I need to the default value of a column in a Content Type. During a SharePoint migration, nothing is more important.

Of course, you could be using [SharePoint migration tools](https://sharegate.com/products/sharegate-desktop) to get the job done. But before you do that, you want to make sure you know what you are copying and where.

During a migration project, it’s the perfect time to [restructure your Information Architecture](https://sharegate.com/blog/sharepoint-migration-preparation). Many of us have learned over time what to do and what not to do, so there is a good chance that you’ll want to organize your Content Types in a similar way in SharePoint 2013.

### The Tools I Used to Get Started

Information Architecture sounds fancy. Sounds like the kind of document you don’t want to oversee. However, it can be a lot simpler than you think, and the best person to do so is the one who already has a decent amount of knowledge on the ins and outs of SharePoint.

Here are some tools that could help you get organized:

* [XMind](http://www.xmind.net/)
* [Mind Manager](https://www.mindjet.com/mindmanager/)
* [Visio](https://products.office.com/en-ca/visio/flowchart-software)
* Book by Ruven Gotz – “[Practical SharePoint 2010 Information Architecture](https://www.amazon.com/Practical-SharePoint-Information-Architecture-Sharepoint/dp/1430241764)”
* [Balsamiq](https://balsamiq.com/)
* Excel
* Word

These are the tools I’ve used to kick start my SharePoint Information Architecture.

A good trick if you are just starting with Information Architecture, try to see what information you need on documents to archive them. This will give you a good starting point to identify document metadata.

In time, you will want to use tools like XMind or Mind Manager to organize this structure quickly. You’ll notice quickly enough, thanks to its easy interface, that many documents are similar in terms of properties or metadata.

You’ll find yourself grouping them together and forming “Content Types”. Once you’re finished, you’ll be ready to create a formal SharePoint Information Architecture.

### It Doesn’t Have to Be Boring

You got into this project to work on SharePoint and now you find yourself having to build an Information Architecture. We’ve all been through it, and at first I felt the same way you are now.

My advice, take it one piece at a time. Go meet the teams and explain what you are doing and why, they won’t want to help unless you can show them how this will be helpful.

Don’t jump right into a 100 pages Word document, open XMind, start organizing some thoughts around it. Believe me; otherwise, you will be going back and forth into this Word document adjusting it constantly.

It can be fun; Information Architecture touches many things, including functionality design.

Navigation of the sites or site collections needs to be mapped and for that, Balsamiq does the trick. If there is a requirement to make an Events Calendar, someone must sit down and use Balsamiq to identify what it will look like. But most importantly, what information will be presented and where is that information pulled from.

## Key Takeaways Before Starting a Migration Project

Although these can be applied at any moment throughout the timeline of your project, SharePoint Governance and Information Architecture are crucial steps in anticipation for a SharePoint migration.

What you need to remember before you start looking at migration tools, is that a solid Governance framework and a complete Information Architecture should be built. Otherwise, you will find yourself moving the same old problems and limitations to the new SharePoint 2013.

Start your Governance plan in pieces, don’t try to attack the whole thing at once, and forget about the complicated 200 document version.

Take it to the next level and make it an Enterprise Wiki. What’s important is to not get caught up in the little details and to really focus on providing a set of rules to help standardize the use, maintenance and customizations of your SharePoint environment.

The Information Architecture helps you identify what can be grouped together, and how it will need to exist in SharePoint. This document could technically be given to a SharePoint specialist outside the company to create the entire solution exactly as it is required.

## **Planning your Migration from SharePoint to Office 365**

you still need to plan and [create a roadmap for your SharePoint migration](#_How_to_Build). But also, a [Governance Plan and an Information Architecture](#_Building_a_SharePoint)will ensure the success of your transition and also check [limitations of SharePoint online](https://docs.microsoft.com/en-us/office365/servicedescriptions/sharepoint-online-service-description/sharepoint-online-limits).

I strongly recommend you start by reading the above articles if all this is new to you. A lot of people tend to see Governance as some kind of out-of-control monster, but with a little patience and planning, you can keep it simple and have it served your needs the way you want it to. This will, in turn, help you manage your SharePoint much more efficiently.

Of course, there is a variable you should be taking into consideration: You're moving to the cloud. So, if you had SharePoint installed on your servers, upgrading to Office 365 means that you may not have the same features available to you.

### Consider Missing Features in SharePoint Online

Even if you're technically migrating from one version of SharePoint to another, they aren’t exactly the same. First, you have to take into consideration that there are different plans you can subscribe to and each with a [set of features available.](https://support.office.com/en-us/article/Business-plans-you-can-switch-to-in-Office-365-7a82c397-a2e2-47ab-b371-39d25a66b582?ui=en-US&rs=en-US&ad=US)

I strongly recommend you do your research and check to make sure the features you were using are still available in your new Office 365 plan.

Of course, this all depends on the migration strategy you've established. I always begin by inventorying my environments, then employing my “RMR Strategy”, which stands for “Remove, Migrate and Rebuild” and helps me identify what to do with each site I have in my collected inventory.

Microsoft has a very detailed article which shows every single feature in a complete [Edition Comparison Chart of SharePoint 2013](http://technet.microsoft.com/en-us/library/sharepoint-online-service-description.aspx), both On-Premises, Online and Office 365.

## Migrate from SharePoint to Office 365 or SharePoint Online – Supported Scenarios

After planning your migration and evaluating the available subscription plans, you’ve identified the content you want to move to Office 365. But how do you proceed?

We've identified 5 different ways to upgrade to Office 365 or SharePoint Online.

* Manually copying files
* Using the Office 365 Migration API
* Using custom coded solutions or a third-party tool
* Microsoft FastTrack
* Hybrid



### Manually Copying Files to Office 365

I included this method because I have to, but it's important to note that, in my opinion, it's the least practical way to migrate.

One way this can be accomplished, is by taking the files using the Explorer View in SharePoint, and moving them manually to the destination. However, by doing it this way, you will lose all metadata as well as the "Created by" and the "Created date". They will be replaced by the person responsible for the copying, at the time he is doing it.

This isn't recommended, because you don't want all your documents to suddenly be owned by one person and all modified at the same date and time.



### Using the Office 365 Migration API

The Office 365 Migration API is a newer way to approach a migration that boosts the speed of migration of files by leveraging Azure.

Essentially, you export your content into a migration package that is sent to Azure Storage. When the timer job runs in Azure, they'll take that package and put the content in your Office 365 environment, based on your package settings. If you're looking for a more in-depth, technical explanation, we've [written extensively on the subject before](https://sharegate.com/blog/office-365-import).

This is a quick solution developed by Microsoft to make your migration easier; however, it’s fairly complicated to set up, and is only for moving content. You’ll need to prepare the entire environment beforehand for it to receive the migration packages.

### Using Custom Solutions or Third-Party Office 365 & SharePoint Online Migration Tools

You could develop your own coded solution to move content over to Office 365, but chances are the time, effort and support that will go into that will be huge when compared to what a third party tool like [Quest](https://www.quest.com/metalogix/), [Sharegate](https://sharegate.com/products/sharegate-desktop), [SharePoint Migration Tool](https://docs.microsoft.com/en-us/sharepointmigration/how-to-use-the-sharepoint-migration-tool) can offer.

It could be worth it to look into a tool that can simplify and accelerate your migration process.

### Data Migration with Microsoft FastTrack

For Office 365 customers with 150 seats or more, Microsoft offers a free data migration service that can also help guide administrators with their move, thanks to tools and other documentation. More information and details on the more technical aspects can be found on [TechNet](https://technet.microsoft.com/en-us/library/mt651702.aspx) and on the FastTrack website.



### Hybrid Upgrade to Office 365

In this scenario, which isn't technically a migration scenario per se, we have both environments running, On-Premises and the cloud. In this Hybrid mode, the environments are linked.

Basically, instead of moving your older content to the Cloud, the idea is to keep running your On-Premises SharePoint and slowly start using Office 365 by creating new Sites there, instead of in the old SharePoint.

After this connection has been made, users will be able to navigate seamlessly between the two, not realizing when they are in one or the other.

This way you can have a more seamless transition as you upgrade or move content to the cloud. Granted it doesn't provide a way to actually move sites from one place to the other, but in some cases, you simply do not need to. Remember the RMR Strategy.

## Migrate to Office 365 on your Own

Migrating to the cloud is a big endeavor and can bring lots of positive changes to the way an organization collaborates. Choosing the right way to get there is probably the most crucial step, so it's important to do extensive research on which method best suits your needs.

The manual migration scenario simply doesn't work for me, as the integrity of the documents isn’t kept. The migration API is a quick and easy way to move your content from a source to a destination, as is the FastTrack method, if you have the right number of seats.

The Hybrid scenario is great and I strongly recommend you use it during your transition, however it provides no real solution to move your content over. Your best bet would definitely be to use a custom solution or a third-party tool that can get everything done with minimal effort.

# Options to analyze existing on-premise environment

[**SMAT**](https://docs.microsoft.com/en-us/sharepointmigration/overview-of-the-sharepoint-migration-assessment-tool)

The SharePoint Migration assessment tool (SMAT) is a simple command line executable that scans the contents of your SharePoint farm to help identify the impact of migrating your server to SharePoint Online with Office 365.

As the tool is designed to run without impacting your environment, you may observe the tool requires one to two days to complete a scan of your environment. During this time, the tool will report progress in the console window. After the scan is complete, you can find output files in the Logs directory. This is where you will find the summary and more detailed insights into the scenarios that could be impacted by migration.

**Inventory Manager File Share:**

FileShare Inventory Tool is used as a pre-migration activity to generate the file and folder inventory for FTC Customers. This tool iterates through all the files and folders shared in available UNCs in the local network.

Tool allows Migrator to list out the number of files/folders, total data size, invalid files/folders and many such information before migration begins.

This tool gives 3 reports which are at File Share-level, file-level, and folder-level.

Based on the report, the migration team can take necessary actions to migrate or exclude the items which are invalid. Customers can use this tool to generate the inventory which will help in identifying the files which will not be migrated, the number of files and folders, the size of each file and other important file/folder attributes.

[**ShareGate Reporting**](https://sharegate.com/blog/new-office-365-groups-reporting-and-more-with-sharegate)

you can use Sharegate to run a report that tells you which of your groups allow external senders, or have auto-subscribed new members to alerts. You can also access all the essential information you might need, like email addresses, names, owners, privacy settings, members, and even the group description. And hey – accessing Quick Actions in your new Office 365 Group reports won’t hurt either!

Custom Scripts

# New and improved features in the SharePoint Migration Tool (SPMT)

### SPMT V3.0.104.3 (Public Preview)

**New features**

The following features were added to the SharePoint Migration Tool Public Preview V3.0.

| **Feature** | **Description** |
| --- | --- |
| Site migration | SharePoint sites that are "out of the box" - sites that do not use any coding or 3rd party tools - can now be migrated. |
| Navigation | Migration of navigation and icons is now supported. |
| Site descriptions | Site description can now be migrated. |
| SharePoint webparts | SPMT now supports the migration of SharePoint webparts. See the full list of SPMT supported web parts: [SPMT Supported SharePoint Webparts](https://docs.microsoft.com/en-us/sharepointmigration/spmt-supported-webparts). |
| Page migration | Only pages in the site asset library can now be migrated. |
| Managed metadata | This release supports the migration of content types and term stores. Global term store migration requires global tenant admin permissions. |

**Improvements**

In addition to several minor fixes, here are the primary improvements made in this release:

| **Issue** | **Fix** |
| --- | --- |
| Stability | General improvements have been made to remove some errors in tool. |

### SPMT V2.1.102.0

**New features**

The following features were added to the SharePoint Migration Tool, Version V2.1.102.0

| **Feature** | **Description** |
| --- | --- |
| Modern design | The SharePoint Migration tool has a new look and feel that is more closely aligned with the Sharepoint Online design for easier use. |

**Improvements**

In addition to several minor fixes, here are the primary improvements made in this release:

| **Issue** | **Fix** |
| --- | --- |
| Stability | General improvements have been made to remove some errors in tool. |

### SPMT V2.1.101.6

**New features**

The following features were added to the SharePoint Migration Tool, Version V2.1.101.6

| **Feature** | **Description** |
| --- | --- |
| Support for CustomGrid list template | The **CustomerGrid** list template (template ID:120) is now supported. The user can now migrate lists that contain a set of list items with a grid-editing view. |
| New PowerShell setting | When using the **Register-SPMTMigration** PowerShell cmdlet, users can now set the parameter MigrateAllFieldsAndContentTypes. |

**Improvements**

In addition to several minor fixes, here are the primary improvements made in this release:

| **Issue** | **Fix** |
| --- | --- |
| Stability | General improvements have been made to remove some errors in tool. |

### SPMT V2.1.101.0

**New features**

The following features were added to the SharePoint Migration Tool, Version 2.1.101.0

| **Feature** | **Description** |
| --- | --- |
| Save session  | The user now has the ability to save their migration session and resume it at later date. |
| Read-only sites supported | The user can migrate a read-only site configured by site policy or the central admin page.  |

**Improvements**

In addition to a number of minor fixes, here are the primary improvements made in this release:

| **Issue** | **Fix** |
| --- | --- |
| Stability  | General improvements have been made to remove some errors in tool.  |
| Reports | The **packageSummary.csv** and **UserNotMapped.csv** reports are now in the details folder.  |

### SPMT V2.1.100.0

**New features**

The following features were added to the SharePoint Migration Tool, Version 2.1.100.0

| **Feature** | **Description** |
| --- | --- |
| Powershell migration solution\*  | All features of the SharePoint Migration Tool (SPMT) can now be done by using PowerShell cmdlets. |
| Settings | Improved labels and text descriptions for settings.  |

**Note:**
To use the SPMT 2.1 PowerShell feature (currently in open beta):

1. Open SPMT v2.1. The PowerShell .dll's will be copied to %userprofile%\Documents\WindowsPowerShell\Modules
2. Run the following:

PowerShellCopy

Import-Module Microsoft.SharePoint.MigrationTool.PowerShell

To learn more, see:
[Migrate to SharePoint Online using PowerShell](https://docs.microsoft.com/en-us/sharepointmigration/overview-spmt-ps-cmdlets)
[SharePoint Migration Tool PowerShell Reference](https://docs.microsoft.com/en-us/powershell/module/spmt)

**Improvements**

In addition to a number of minor fixes, here are the primary improvements made in this release:

| **Issue** | **Fix** |
| --- | --- |
| Stability  | General improvements have been made to remove some errors in tool.  |
| Permissions settings | Separate settings are now available to set file share permissions and the SharePoint on-premises permissions.  |
| Changes to migrating multiple versions  | Checked-in version(s) of a file will migrate but any the checked-out version will not.  |

### SPMT V1.1.90.1

**New features**

The following features were added to the SharePoint Migration Tool, Version 1.1.901.

| **Feature** | **Description** |
| --- | --- |
| Allow migration of 0 bytes files  | Files will be migrated even if they are of zero bytes.  |
| Computer names column  | A column containing the name of the computers running the migration job has been added to the report.  |
| Support of incremental check on target environment  | In SharePoint Online, an incremental check of the target environment will be performed. If the modified time of the source file is earlier than the modified time of the target file, the file will not be migrated.  |

**Improvements**

In addition to a number of minor fixes, here are the primary improvements made in this release:

| **Issue** | **Fix** |
| --- | --- |
| Stability  | General improvements have been made to remove some errors in tool.  |
| Permissions fixes  | We have made several improvements to better preserve the permission when requested and not removing existing permission in the destination.  |
| Warnings when files are checked out  | Users will now have warning messages appear in the tool when attempting to migrate a file that was checked out. |
| Report when performing only a scan  | The **FilesReport.csv** file will now show the correct results when only scanning option is turned on.  |

# Best practices for improving SharePoint and OneDrive migration performance

Migration performance can be impacted by network infrastructure, file size, migration time, and throttling. Understanding these will help you plan and maximize the efficiency of your migration.

Currently, Microsoft's [SharePoint Migration Tool (SPMT)](https://docs.microsoft.com/en-us/sharepointmigration/introducing-the-sharepoint-migration-tool) as well as several third party vendor tools utilize the SharePoint API for migration. It leverages Azure and uses channels for large content transfer. Regardless of which migration tool you use, these factors will apply. Follow the recommendations listed below for each phase of your migration process.

## **Before migration**

Planning is the key to optimizing your migration. Determine what content you need to migrate, prioritize when the content needs to be migrated, and decide on what the optimal migration infrastructure should be.

### I. Scan the source

The first rule of a good migration is to know your source; evaluate and triage your content before you migrate. What content really needs be migrated? What can be left behind? How many file versions should be included? The amount of content you migrate will determine the overall size of your project.

### II. Package the content

This step is where the tool creates a proper package for the content to be imported into the cloud. This step is automated in SPMT and in most third-party tools.

**Package size.** To improve migration throughput, we recommend that you package at least 250 files per transfer. For the transfer size we recommend a minimum of 100MB and less than 250MB per package. This will result in a faster upload speed to Azure and leverages the scale capabilities of the migration API.

The following table provides estimates of the type of speed you may achieve based on the types of content you are migrating.

| **Type of metadata** | **Examples** | **Maximum** |
| --- | --- | --- |
| Light | ISO files, video files | 2 TB/day |
| Medium | List items, Office files (~1.5MB) | 1 TB/day |
| Heavy | List items with custom columns, small files (~50kb) | 250 GB /day |

* Large file size migrates faster than smaller ones. Small file size can result in larger overhead and processing time which directly impacts the performance.
* Files migrate faster than objects and list items.

The speed of this step depends on the efficiency of the tool you are using and the type of content that you package. Splitting your packages in a smart way is something that will greatly improve this step. In addition, ensure that your permissions, sharing, or other limits are set up properly for migration and are within [SharePoint Online limits and boundaries](https://docs.microsoft.com/en-us/office365/servicedescriptions/sharepoint-online-service-description/sharepoint-online-limits).

## During migration

### I. Upload to Azure

SPMT or your third-party tool will migrate your content into SharePoint Online using the Migration API, leveraging Azure as a temporary holding place.

If you have a good connection and can configure your datacenter, choose the same datacenter location closest geographically to you for your Azure and your Office 365 account. Migration data throughput is highest during off-peak hours, which are typically nights and weekends in your region's time zone. Your region's time zone is determined by where your SharePoint Online tenant is set up.

### II. The Migration API

The final step of the migration process is when the data is moved from Azure to SharePoint Online. This action is transparent to the user when using SPMT or a third- party tool.

To improve throughput, users are encouraged to run parallel tasks against different site collections if possible. We recommend that you do not submit more than 5,000 migration jobs/requests at one time. Over-queuing the network will create an extra load on the database and slow migration down. Make sure your task has completed before you upload a new migration request. Some tools may already be doing this for you.

During migration, it is not uncommon for your migration task to be throttled. Throttling is implemented to ensure the best user experience and reliability of SharePoint Online. It is primarily used to load balance the database and can occur if you misconfigure migration settings, such as migrating all your content in a single task or attempting to migrate during peak hours.

For more technical background and information, please see

* [Migration API Overview](https://docs.microsoft.com/en-us/sharepoint/dev/apis/migration-api-overview)
* [Avoid getting throttled or blocked in SharePoint Online](http://go.microsoft.com/fwlink/?LinkID=619858&clcid=0x409)

## **After migration**

After the migration is completed, verify that your content has been successfully moved to SharePoint Online or OneDrive.

## FAQ and Troubleshooting

Question: My migration is going so slow or I am being throttled. What can I do?
Answer: Check that you have configured your migration settings properly. Turn off any software that you do not need to use during migration. For example, disable any file synchronization program or antivirus program on the migrated content. This will help reduce throttle and improve performance.

Question: I continually getting throttled while I am attempting to migrate. Can Microsoft turn off the throttle to help me with migration?
Answer: Unfortunately, we are not able to disable throttle. Throttle is built into our server to protect the database from going down. If you are being throttled with a 429 error, it's an indication that your migration tool is overly aggressive and is over its allocated quota. Please try to migrate during off-peak hours or reduce the number of VMs you are using.

If after several days you are still experiencing excessive throttling, please open a Microsoft support ticket. Include the following in your support ticket:

* How often are you seeing the throttle (e.g. throttle count/hour)
* How much data were you being able to migrate (e.g., 2MB per hour or per day)
* The name of the third party app are you running
* The total size of the content you wish to migrate
* Your migration schedule
* Your Company name and Tenant URL

Question: How much can I migrate per day?
Answer: Plan to migrate at a maximum of 2TB/day.

Question: I have a very big migration (> 100 TB) and I would like some help, who should I contact?
Answer: For larger than a 100TB migration, please submit a support request with Microsoft indicating that you are doing a large migration (>100TB). Follow these steps:

1. Click on **Need help?**
2. For the title, enter **"SharePoint Migration over 100TB"**.
3. Include all of the following on the support ticket:
	* Your company name and Tenant URL
	* Estimated size of your migration
	* An estimate of when you would like to start and complete your migration
	* Describe where you are migrating your content from, such as SharePoint Server, Box, GDrive, File shares, etc.

Question: I have tried everything, but nothing works. Now what do I do?
Answer: Open a support ticket with [http://support.microsoft.com](http://support.microsoft.com/).

 **Reference**:

* <https://rencore.com/blog/5-reasons-to-avoid-full-trust-solutions-in-sharepoint/>
* <https://docs.microsoft.com/en-us/sharepointmigration/migration-assessment-scan-full-trust-component-results>
* <https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/sandbox-solution-transformation-guidance>
* <https://docs.microsoft.com/en-us/sharepoint/dev/sp-add-ins/get-started-creating-provider-hosted-sharepoint-add-ins>
* <https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/remote-timer-jobs-sharepoint-add-in>
* <https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/create-remote-timer-jobs-in-sharepoint>