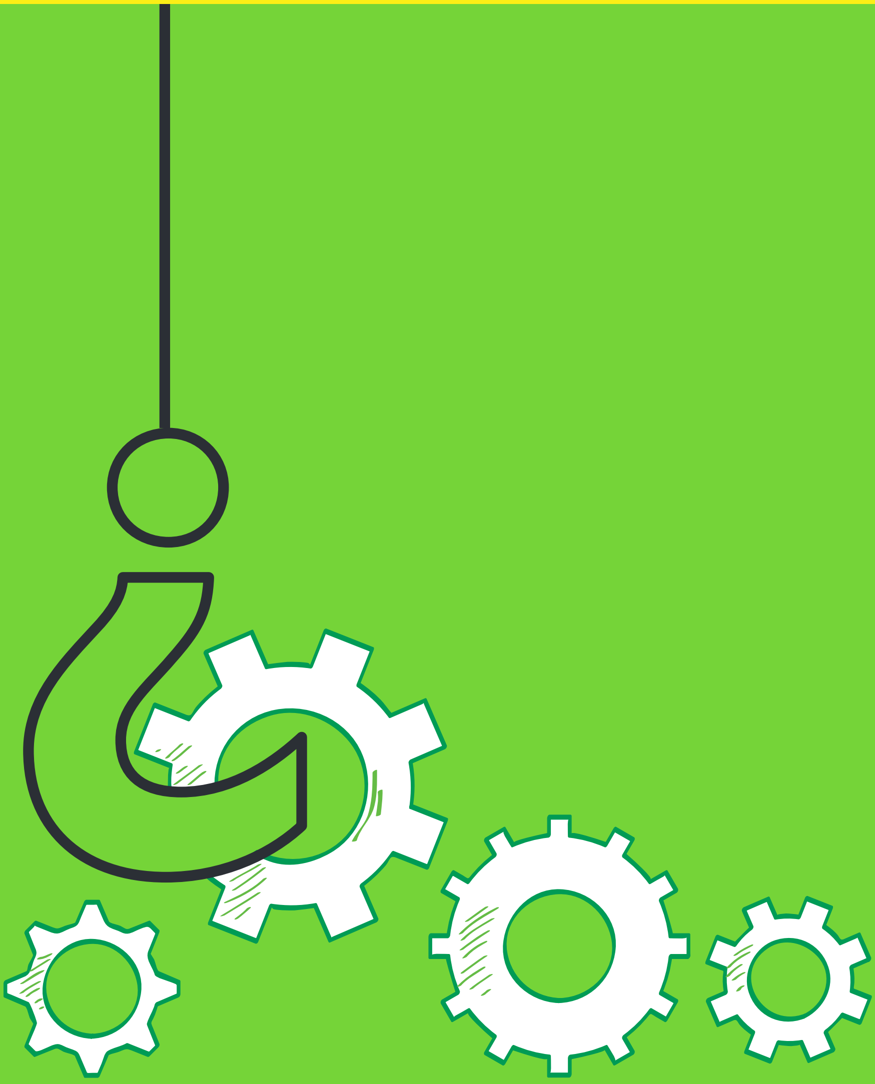
This may be out of date. Read what's new here!



HOW TO CHOOSE THE RIGHT ASP.NET CONTROLS FOR YOU

Telerik

CONTENTS

About the Author

Jeffrey T. Fritz is a Microsoft MVP in ASP.NET, an ASPInsider and Developer Evangelist for Telerik with over 15 years of experience building large-scale multi-tenant web applications in the software-as-a-service model. A Penn State graduate with a degree in Management Sciences and Information Systems, he speaks regularly at developer events through the INETA speaker program and maintains blogs at www.csharpfritz.com and blogs.telerik.com/jefffritz. You can find him on Twitter at @csharpfritz and can reach him at jeff.fritz@telerik.com.

Several years ago my team and I were building and maintaining the UI of our application ourselves. We would get annoyed, frustrated, and fed up every time a new device or browser version would break it. We would spend days making that section of HTML and JavaScript work just right and instead of solving business problems, we were fighting a user-interface issue.

At that point, one of the project supervisors tasked me with reviewing the available third-party ASP.NET controls on the market and recommending the best option for our project.

If you too are currently evaluating products to help you deliver a better product on time with less worry about the user-interface issues, then this guide is exactly what you need. See what I learned from that task and in working with many third-party tools since that time.

My goal is to give you a checklist of things to look for to ensure you have the best fit for your projects. At the end of the guide, you will find a link to a ready-to-use spreadsheet where you can quickly rate the different solutions on predefined factors.

PRO TIP:

Narrow your list. Thoroughly evaluate two to three options max.

Start by searching out and reading reviews from other publications about the products. Talk to your peers in your area and get their opinions.

Use these inputs to help narrow the field to two or three options that you can work with. Building a proof of concept and reviewing features of four or more products is a nightmare. That many products in a review can also be overwhelming for your organization to understand your findings.

Factor 1:

FUNCTIONALITY

When reviewing functionality, I consider the following:

- What do I need to build my project right now?
- What will I need to add to my project in the next six months? Does this product have something that can assist in that future scenario?
- What did I build recently that I can replace with this product and make better for both the developer and the website visitor?

When making your initial review of the products, you need a quick way to determine if they meet your needs.

1. Review online and downloadable demos of the product.
    a. Are there scenarios that match or come close to what you are attempting to achieve?
    b. Is there source code provided for these demos? Does it look like something you could easily re-implement in your project?
2. Review the product documentation and see if it is something you can work with.
3. Put together your project requirements in a concise document and send them to the vendor for assistance.

All of the commercial vendors want you to succeed and will do their best to help you understand how their product can meet your needs. An engaged vendor will email you back and possibly start a phone call to discuss your needs and demonstrate how their product addresses your project.

Factor 2:

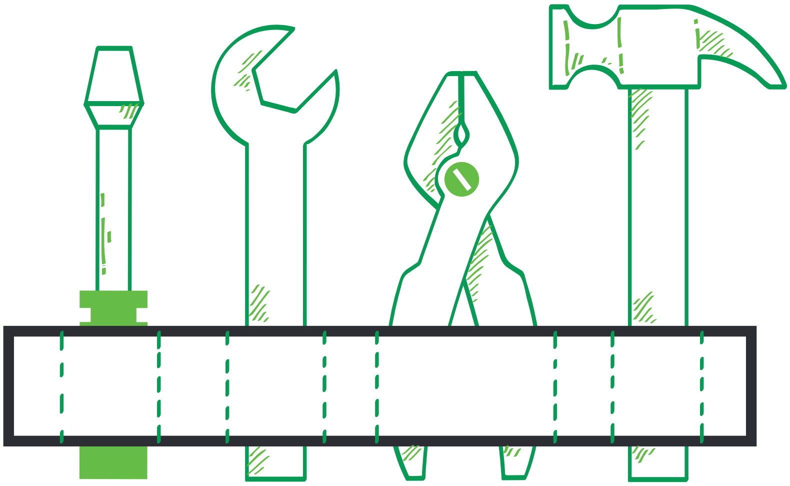
EASE OF USE

You are a trained software developer; do you really want to spend more time training to learn to use a new product? Of course not. Look for a product that does not have a steep learning curve. You should be able to find a good web tool that has the following properties:

- Uses familiar frameworks like AJAX or jQuery
- Comes with Visual Studio Design-Time Wizards for codeless configuration
- Features intuitive API that you can learn to use and apply uniformly across controls in the toolset
- Has rich documentation and learning resources—where can you go to learn more? If you want to learn more advanced techniques with the product, is there a resource available?

Not every use of the product will require advanced, in-depth use of most of the product API. You should be able to accomplish common tasks with little to no effort, yet have the flexibility to take a deep-dive and make the product do some amazing things with some effort.

Factor 3:
CUSTOMIZABILITY



This is a great factor that I had fun exploring. How much can you get out of the product by changing it? If you had access to the source code of the product, how easy would it be to extend and add features to it of your own design?

This may not be something that you need to focus on, particularly if producing software is not the primary revenue generator of your organization. However, if you are producing software to sell, you will want to be able to modify and extend the tools to be uniquely suited and branded for your organization.

This is why, if you are a paying customer, you should want to get access to source code for the tools you are working with.

Factor 4:

PERFORMANCE

I have a friend who claims that performance is the number-one feature of any product that he delivers. I can agree with this perspective and I think you should consider it when you review ASP.NET tools. The average user is becoming less and less patient with the performance of the web.

The rule of thumb is that a visitor should not wait two seconds for a page to load. This rule becomes very questionable when we start to take into account smaller devices with less memory and processing power. Add on top of that questionable connectivity to a wireless network and you quickly run into scenarios where the web site you are providing needs to be downloaded and available on screen quickly.

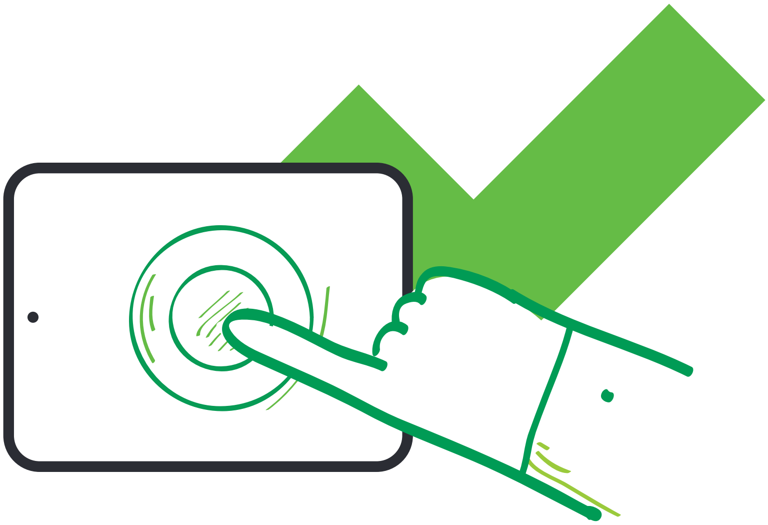
Consider these points when thinking about performance with web controls:

- How much markup is generated?
- Can the JavaScript/CSS/static supporting content be retrieved from a content delivery network closer to my end users?
- How quickly is the markup generated?
- How is the performance of JavaScript executing on the browser for these controls?

Take this performance review and pair it with the demos that you are looking for in the first factor and you should be able to evaluate performance without having to build out a performance testing environment.

Factor 5:

TOUCH AND MOBILE SUPPORT



Welcome to the future. The web is no longer about just keyboards and pointers, and there are almost as many touch-enabled devices using the web as there are non-touch enabled devices. If the trends continue, you can expect that touch will become the dominant interaction with the web in the next three years.

Keep this in mind when you choose third-party ASP.NET controls. You will want to be able to support those tablets and phones that are using your application. Look for the following in a control set:

- Support for larger 'touch regions' that your fingers can easily find without interacting with other buttons and widgets nearby
- Gesture support that enables you to swipe, flick or touch and hold on items to interact with them—remember, touch devices do not have the ability to hover as a pointing device can
- Alternate renderings to support smaller devices—can the controls format content on a phone or tablet in a compelling way without dominating the viewport?

I used to think of this as "planning for the future." Then I looked around the conference room during a team meeting and noticed that EVERYONE had tablets and smart phones. That future is now, and we need to consider this factor a higher priority than "sometime after the desktop version of the site works."

With the "bring your own device" craze, this is important even if you're building internal applications.

Factor 6:

SECURITY

Does the NSA love the product you are reviewing? Do they like it for all of the WRONG reasons?

Do some checking to see if there are security questions around the tool. Some example questions may include:

- Does it encrypt data as it is transmitted between systems?
- How does it store and handle authentication credentials?

Ask some direct questions of the vendor. If you have any concerns with presenting sensitive information, you should consider security a major factor when choosing a control set.

Additionally, how responsive will the vendor be if a security issue is discovered? Do they have mechanisms in place to issue updates? Can you get an intermediary patch to correct any security holes that are exposed in their product?

Security is like air: most people don't care about it until they don't have any. Make sure you choose a secure product first, and let someone else work on protecting your user-interface components.

Factor 7:
ROADMAP

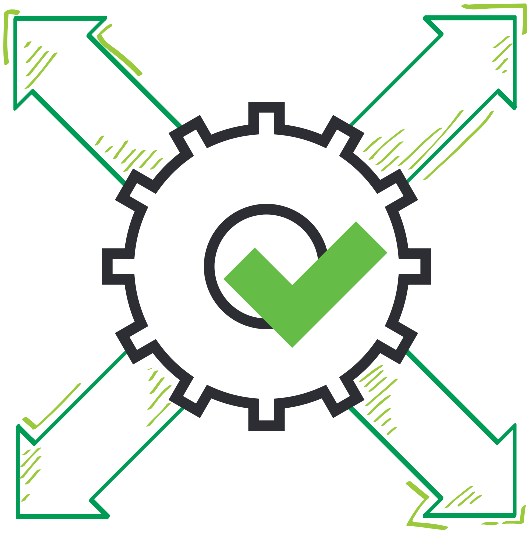
Are you planning to use the product for the next two or three years? Is the vendor planning on building and supporting it for the next two or three years? Is the product "done" active development by the vendor? Will you receive any more cool features or support for new devices and browsers, when you are paying for maintenance releases?

A good vendor with a product that has a significant future for them will publish a public roadmap on their website. This should include items about adding features and maintaining the existing set of features across new browsers and devices. The roadmap should also include information about how frequently they plan to issue these updates and how you can expect to receive those updates.

Telerik makes a clear roadmap for the UI controls for ASP.NET AJAX available on their website at all times. Not everything that will be delivered in each new release is listed, as they need to keep SOME secrets and surprise new features to themselves until the product is released. It also helps to keep the competition honest in what they are developing. You will always be notified when a new version is ready for download through the Telerik Control Panel and plugins in Visual Studio.

Factor 8:

ACCESSIBILITY

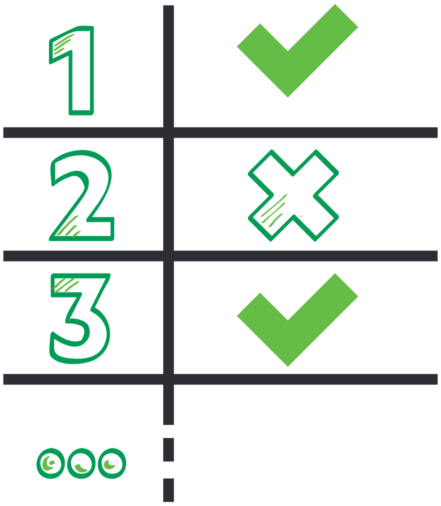


I'm not as young as I was when I started in this profession and I am starting to increase the font sizes in my text editors and word processors so that I can see the text I am writing. Will you have similar problems with the controls that you are considering? Are they fixed at one size and cannot adapt to present content that is accessible to everyone?

Can the controls be localized? What happens if I need to use the controls to present information in English, Spanish, Russian and Chinese? How will they handle those unique requirements?

If you do not have a need for internationalization, you should consider the needs of your older users and those who could have a problem interacting with your website. A great control set will have support for WAI (Web Accessibility Initiative) and ARIA (Accessible Rich Internet Applications) standards.

THE ULTIMATE EVALUATION WORKSHEET



These eight factors are the most important options that I look for when I am evaluating controls or libraries of visual components to use with my web applications. I have put together a brief worksheet that you can use to help evaluate your findings as you review the various options to meet your project's unique needs.

No one will tell you what the correct product is for your project, but can certainly point you in the right direction to provide answers to some of your questions. Use the worksheet to help determine the best option for you and if you have any questions, ask me on Twitter. You can find me at twitter.com/csharpfritz.

Get the evaluation sheet