One common hurdle every developer encounters at some point is the infamous browser discrepancy issue. You may have made your website flawless in your go-to browser, but as soon as you launch it, there's always someone who tells you "it looks off" on their browser. It's a headache that pops up because different browsers interpret certain aspects of your code differently, and this is where cross-browser testing steps in to save the day.

**What is cross-browser testing?**

In essence, cross-browser testing is about ensuring that your website works seamlessly across multiple browsers. This means that it looks good and functions as expected, no matter if your user is browsing on Chrome, Firefox, Safari, Opera, or even the good old Internet Explorer.

**Why should I do it?**

Cross-browser testing is all about user experience. Your users aren't all using the same browser, or the same version of the browser, or even the same device. If your website looks great on Chrome but falls apart on Safari, you're not providing a consistently good experience for all your users. Issues like these can knock down your website's credibility, so it's best to sort them out before launch.

**Who should do it?**

If you're in the web development game, cross-browser testing is part of the job. Whether you're a developer, a QA professional, or a UX designer, ensuring cross-browser compatibility is crucial. Usually, it's the developer who bears the brunt of the responsibility to ensure the website behaves well across browsers.

**How is it done?**

1. **Identify your targets**: First, you need to figure out which browsers and devices your audience uses. There's no point testing for a browser that none of your users are on.
2. **Prioritize**: After you've got your list of target browsers and devices, you need to prioritize. Start testing on the browsers that most of your audience uses.
3. **Test**: Check your website's visual elements, functionality, performance, and responsiveness on each browser. Look for any glitches or inconsistencies and note them down.
4. **Debug and fix**: Once you've identified the issues, it's time to dive back into your code and fix them.

**What tools can I use?**

Luckily, there's a heap of tools out there to help with cross-browser testing. These include automated testing tools like Selenium and TestComplete, cloud-based testing platforms like BrowserStack and Sauce Labs, and even some browser-specific developer tools.

Cross-browser testing may can be a bit of a chore, but it's an essential part of web development. It's all about ensuring that all your users get the same high-quality experience, no matter what browser or device they're on. Taking the time to get it right the first time around will prevent it becoming an even bigger headache later.