Usability and Accessibility: What every Tester should know

by Stamford Interactive

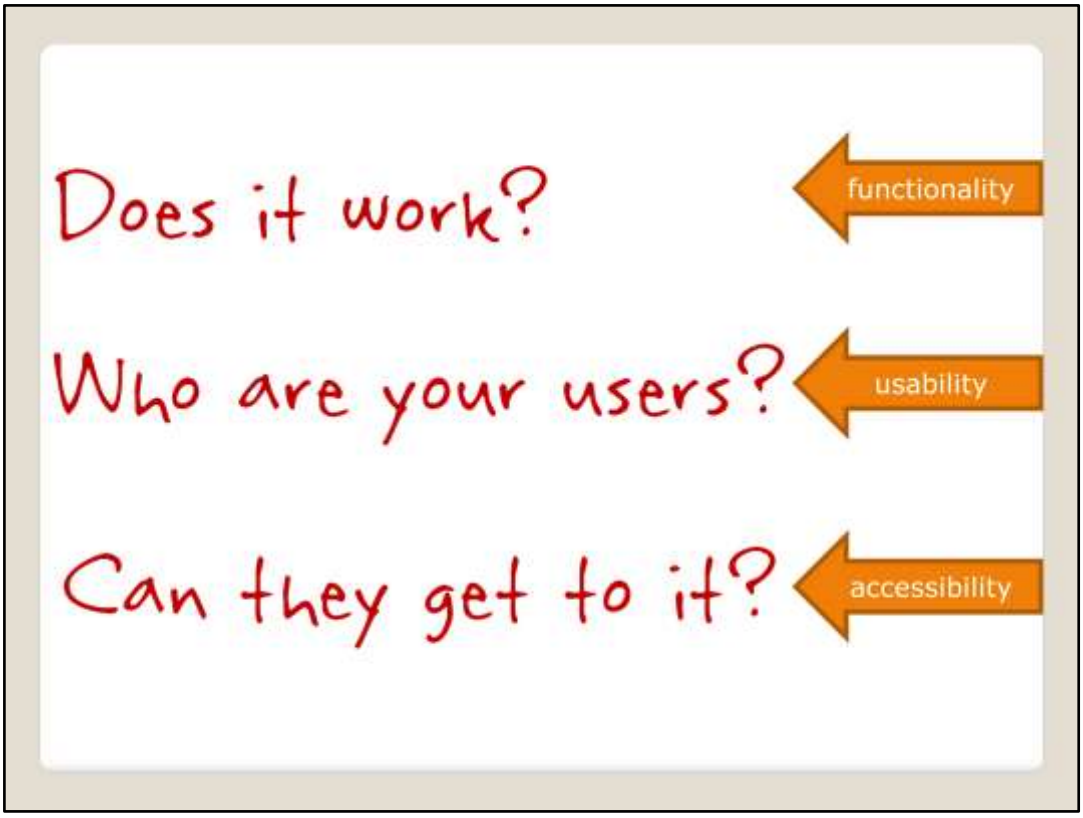
www.stamfordinteractive.com.au

People often ask us upfront if a site or product is accessible. Or if it's usable. In both cases, despite the official standards, the usual answer is that "it depends".

As testers, you are the last bastion of quality control before the system or product is thrust upon a user.

My task today is to give you the questions you need to be asking.

It's important that you're considering these questions, because the people up the production line might have missed them.

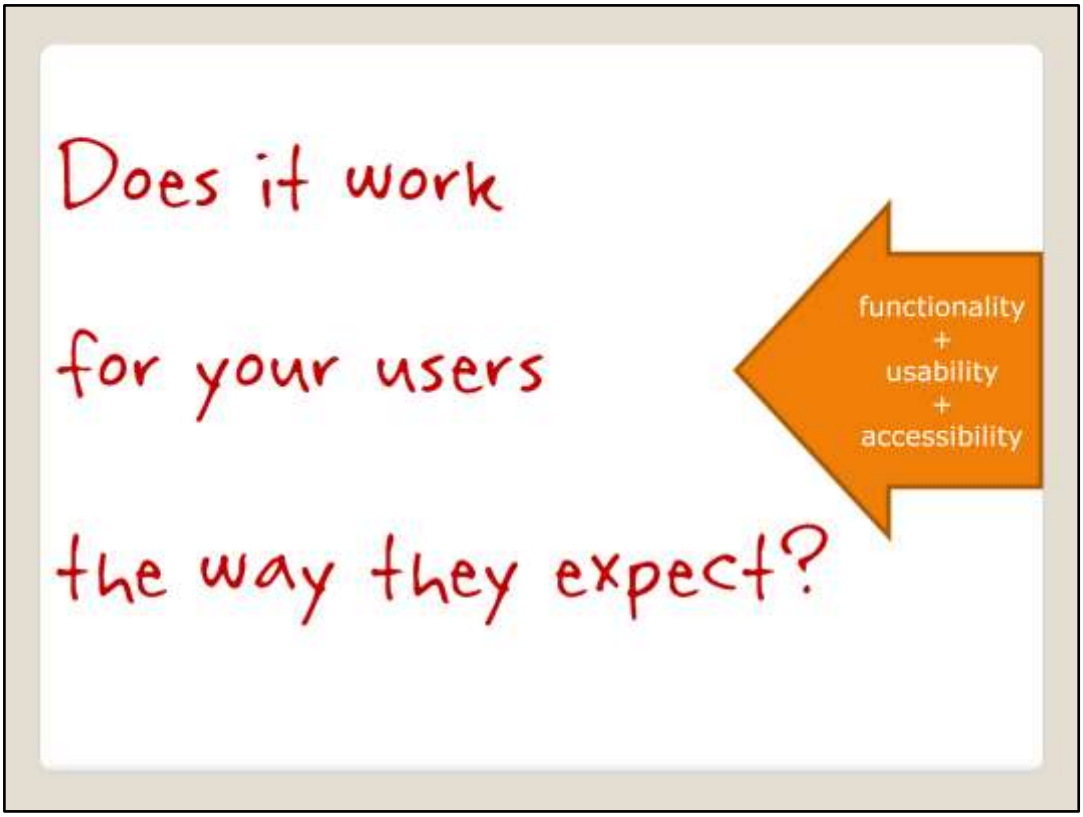


There are 3 main questions here that each area seeks to answer:

- Functionality: Does it work?
- Usability: Who are your users?
- Accessibility: Can they get to it?

Each focuses on different aspects of testing, and is often looked at in isolation from the others.

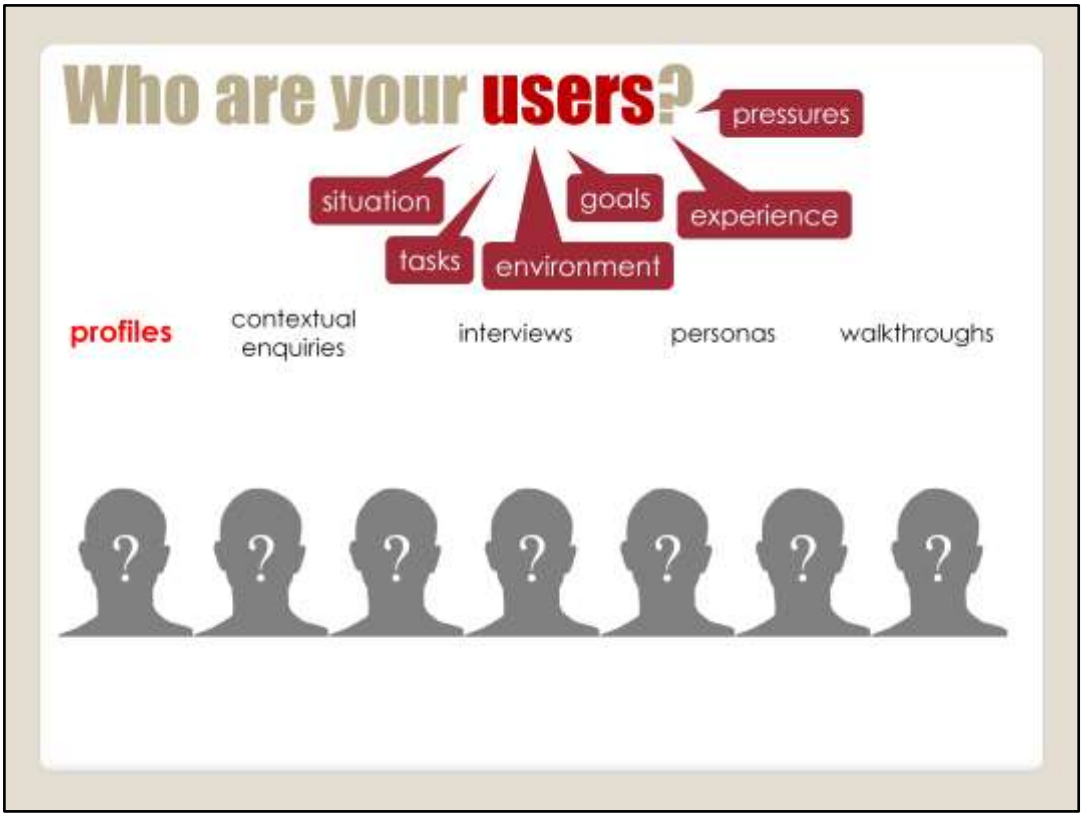
But put them all together, and you actually have a complex question...



... that then demands that you understand *who* your user might be, what their situation is, what their preferences are (behaviours, norms, background), and *then* think about how they can perform their task. Then it becomes a matter of how you can cater for all those needs.

In the end, it simply boils down to being considerate.

Like that dinner party you threw where you managed to cater for your friend who's on a complicated diet, her vegan boyfriend, your mate who is allergic to chilli, and the child who won't eat anything except chips.

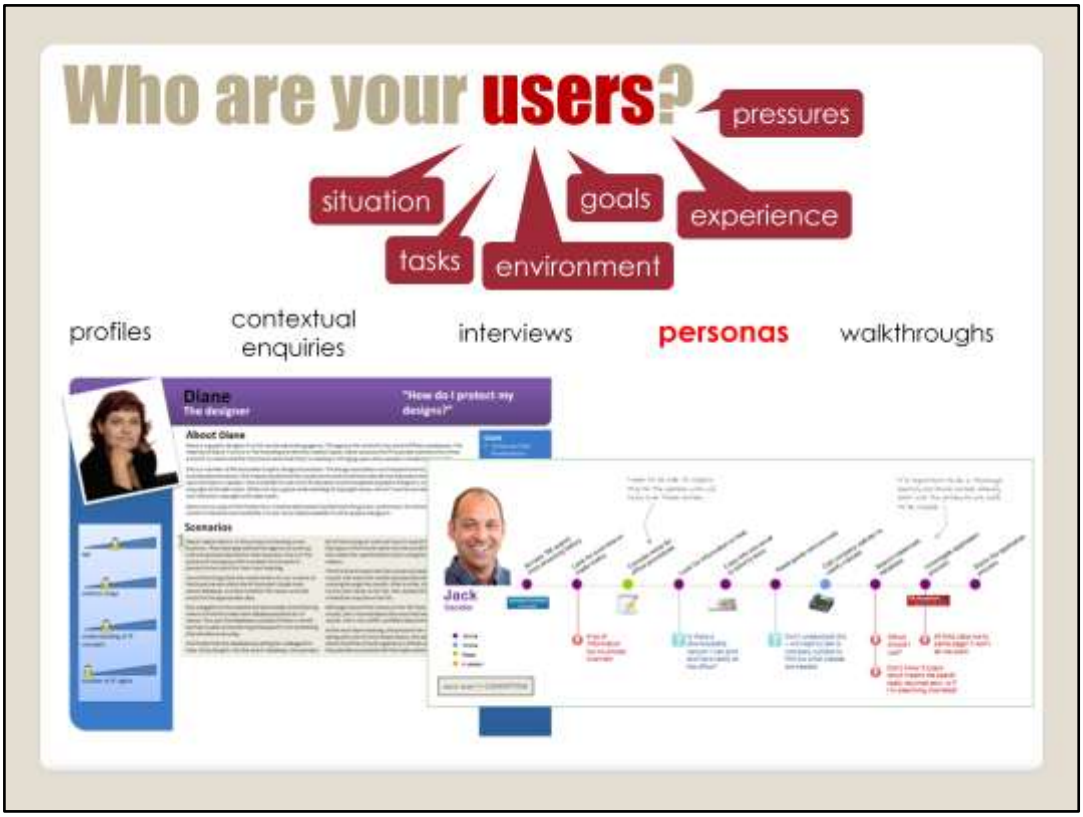


So who *are* your users? There are number of techniques we use in the user experience field that can help us learn more.

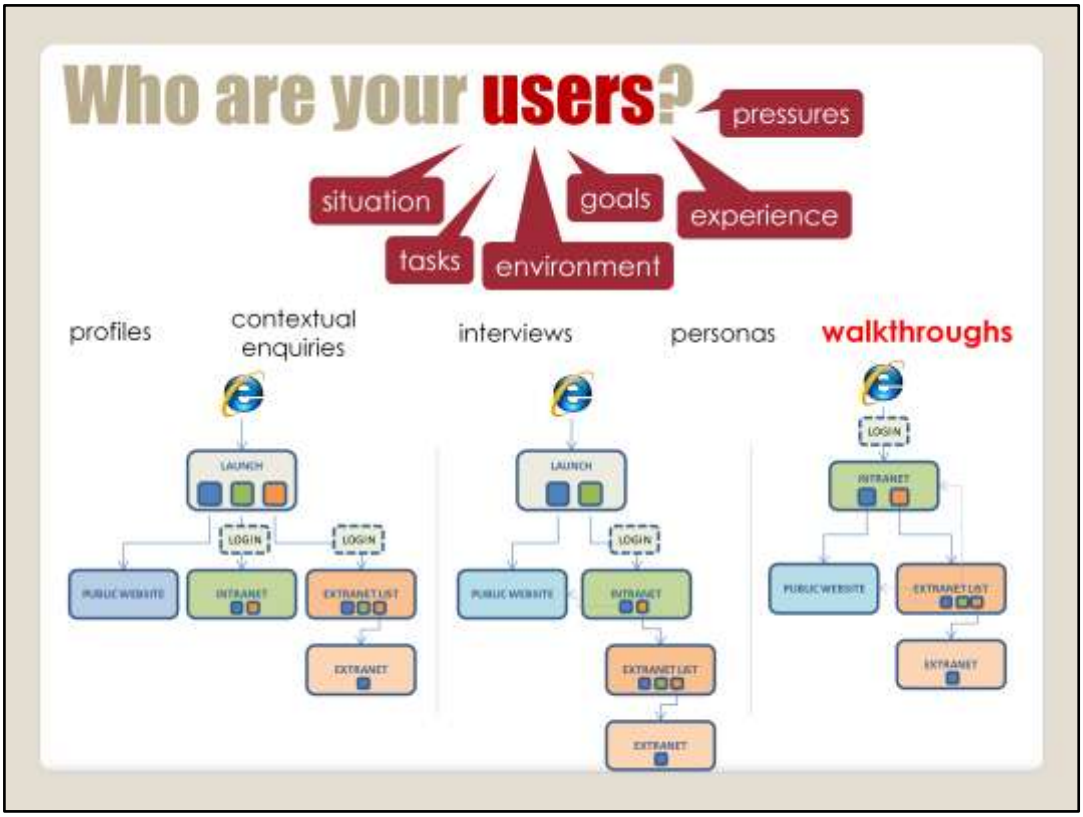
Profiles help you identify who your intended/existing users will be. Target them, test with them. If your designers have already identified profiles of users, get your hands on the profiles and make sure that you test from their perspectives.

Contextual enquiries and **interviews** place you in users' footsteps. Watch them and see what they actually do.

The experience of waiting in a queue or dealing with impatient customers can directly influence how you design software for a reception desk. Understanding what pilots have to process affects how you design instrumentation. Finding out what "quick help" call centre staff place around them gives you a clue as to what they need to do quickly.



Personas are fleshed-out character representations of your profiles, and are based on research from contextual enquiries and interviews. They help keep the users in the foreground. Again, if your designers already have these, get hold of them and test from their perspectives.



Walkthroughs are the best way to test options early, so you can keep refining designs and processes. No need to be formal – just do it as early as possible. If you have a number of options you're considering, test them out. Paper prototypes, guerrilla testing, asking super users – these are all valid ways to refine designs.

For instance if you have software that you're updating, and you have a helpdesk, run your ideas past them. They will immediately tell you what works and what doesn't, because they are the ones who have to walk people through it and they know your users.

So now that you have an idea of who the target users are, how much do you know about how they operate? What influences their behaviour? What tools do they have? How proficient are they?

What preferences and devices might your users **want** to use? Are they mobile? Like a particular browser? Have you therefore tested with all of them?

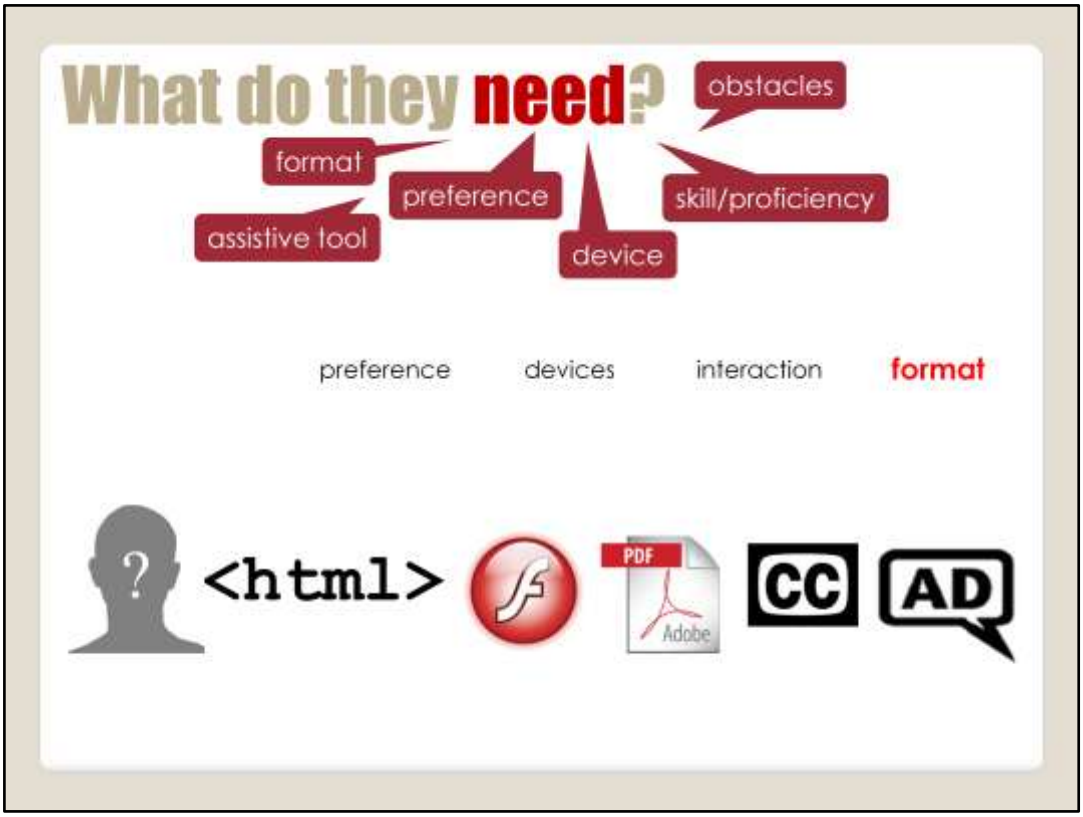
What about what tools the user **has** to use?

This also includes the person who:

- Wants to check in for his flight, but only has his mobile
- Forgot his mouse, and his laptop trackpad is malfunctioning
- Eyesight not what it used to be, hates all this grey on grey design that makes it hard to read

This also includes the person who might be:

- Interacting via a keyboard, or other input device such as mouth sticks, eye tracking, dictation...
- Listening to text via screen readers
- Requiring a different format such as sign language (AusLan)
- Reading through Braille output



And what about the format the user wants their content in?

This includes the person who might be:
- Requiring a different format such transcripts, or captions, or audio descriptions
- Can't process PDFs
- Is on a lower bandwidth and so needs a light version of content

So there's a lot to know about who your users are, and what they want. Now some of you might ask – all this effort for a few?
Let's shift that focus. Instead of thinking about it as designing for the needs of the few, think about designing for **universal access**.

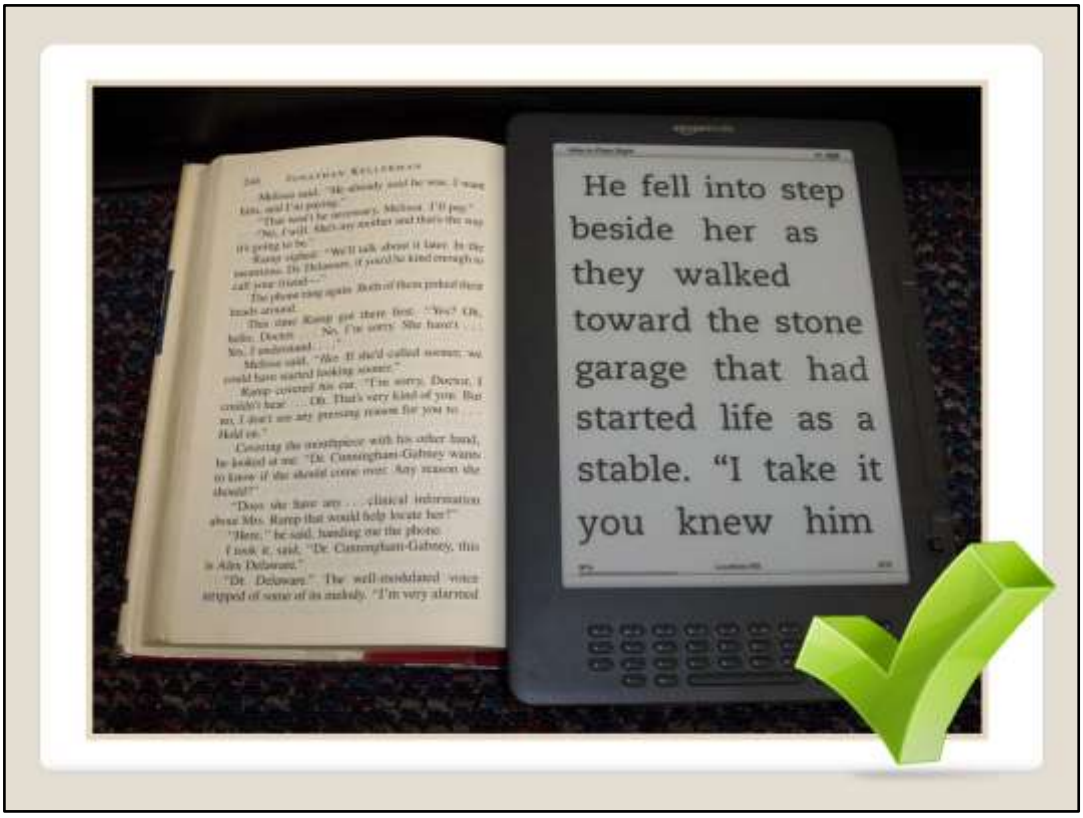
Here are some beautiful examples of how systems can be designed to let users just "get on with it".

The audio cues at the traffic crossings are now a standard cue for everyone that they can cross. People can afford to shift their attention to other things – conversation, looking elsewhere, being on the phone, etc. As soon as the audio cue is heard, we all know we can start walking. Including the blind pedestrians it was initially designed for.

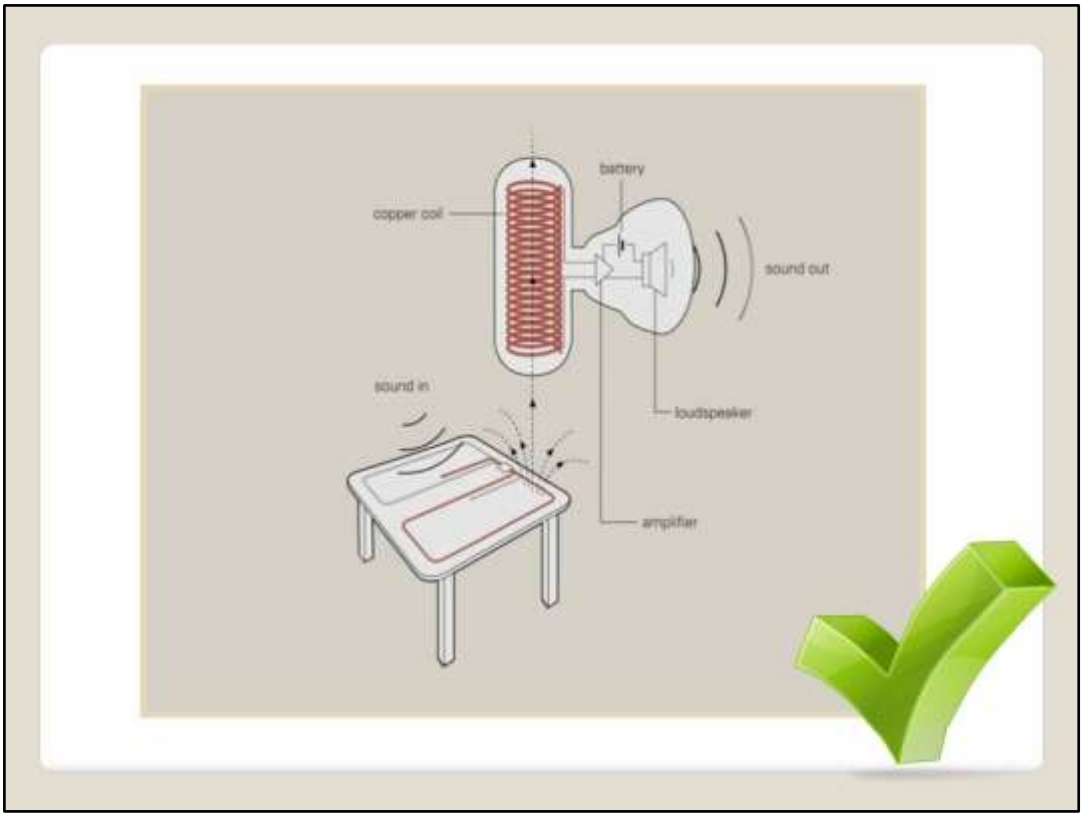
These measuring jugs are lightweight, with a large grip and a tilted measure that allows you to look down into it and still measure accurately. No need to raise to eye level to check. Its clever design makes it appealing to everyone, including the group it was initially designed for: people with mobility problems, such as arthritis.

Ramps are a relief for people with wheels – luggage, trolleys, skates, prams – and yes, the original audience – wheelchairs.

Adam Hills included a sign language interpreter in his comedy acts, which added another level of humour (especially for his crude jokes) as well as making the content understandable to the deaf community.



The Kindle is not just a portable library, it also provides options for the reader’s comfort. Without having to use any other device to enlarge text (glasses, lenses), the Kindle comes with built-in magnification. This is life-changing for people with deteriorating eyesight who might have lost access to their world of books.

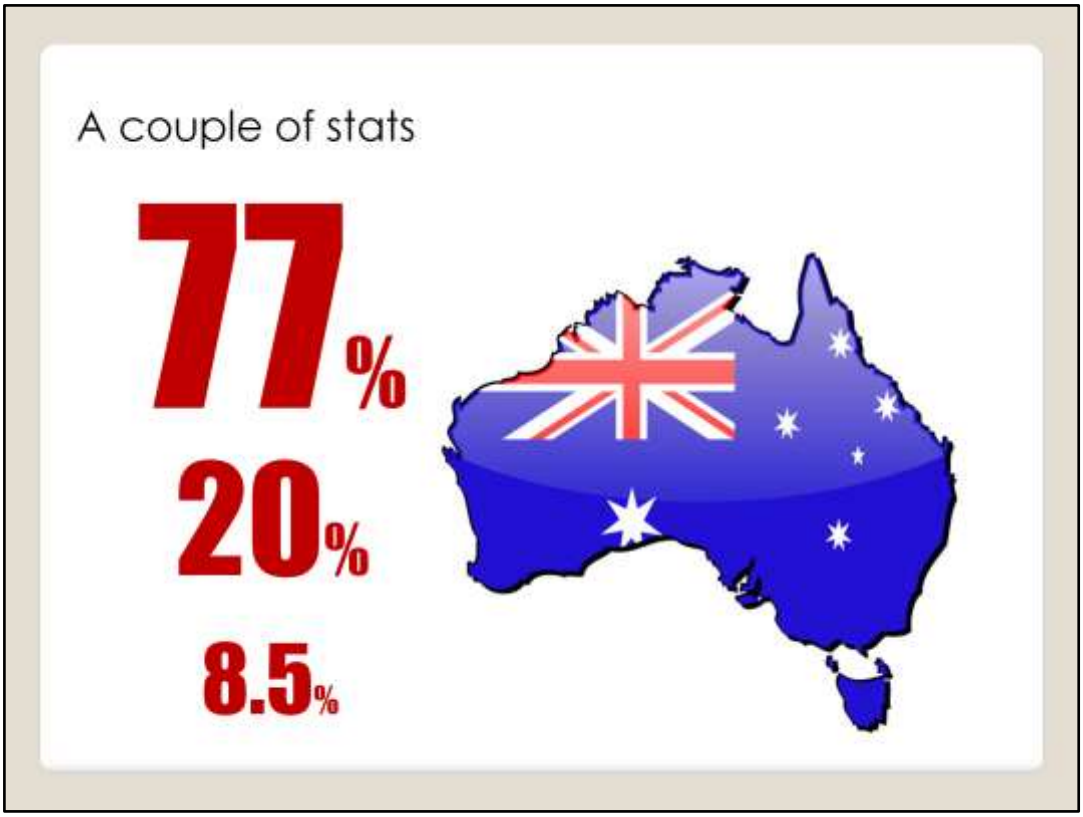


Design company IDEO's "TableTalk" concept demonstrates using the hearing loop as a great example of inclusive design. The earpiece picks up any sound that is amplified by the loop around the table. Originally designed for the hard-of-heading in facilities such as theatres and auditoriums, this is something everyone could benefit from at the noisy local pub. With everyone talking regularly around the table, even with the noisiest background, the conversation is clear and unstrained.

A recent TED talk by Dennis Hong (http://www.ted.com/talks/dennis_hong_making_a_car_for_blind_drivers.html) described a project to make a car for blind drivers. Not as passengers, but as decision-making, independent members of society who could partake in activities the rest of society takes for granted. The beauty of this is that the technology from the project will seep into mainstream technology, and I personally can't wait for sensors to help me parallel-park. Another great example of design that will benefit everyone.

Do you recall rushing for your flight and needing a quick way to check in, but couldn't get to an internet connection or a printer? The mobile app is a beautiful example of a solution that fits the situation.

Accessibility isn't just for addressing disabilities or impairments. It's about "access". And that applies to everyone.



But because disability is a driving force for addressing accessibility, it's important to understand why.

77% of people have a **chronic condition** in Australia (ABS 2006) while **1 in 5** suffer a disability. The 20% figure is fairly consistent amongst developed countries, but may be higher in underdeveloped countries. An ageing population ensures that this figure is set to rise with rates of disability up to 41% of those aged 65-69 years, and up to 81% of those aged 85 years and over.

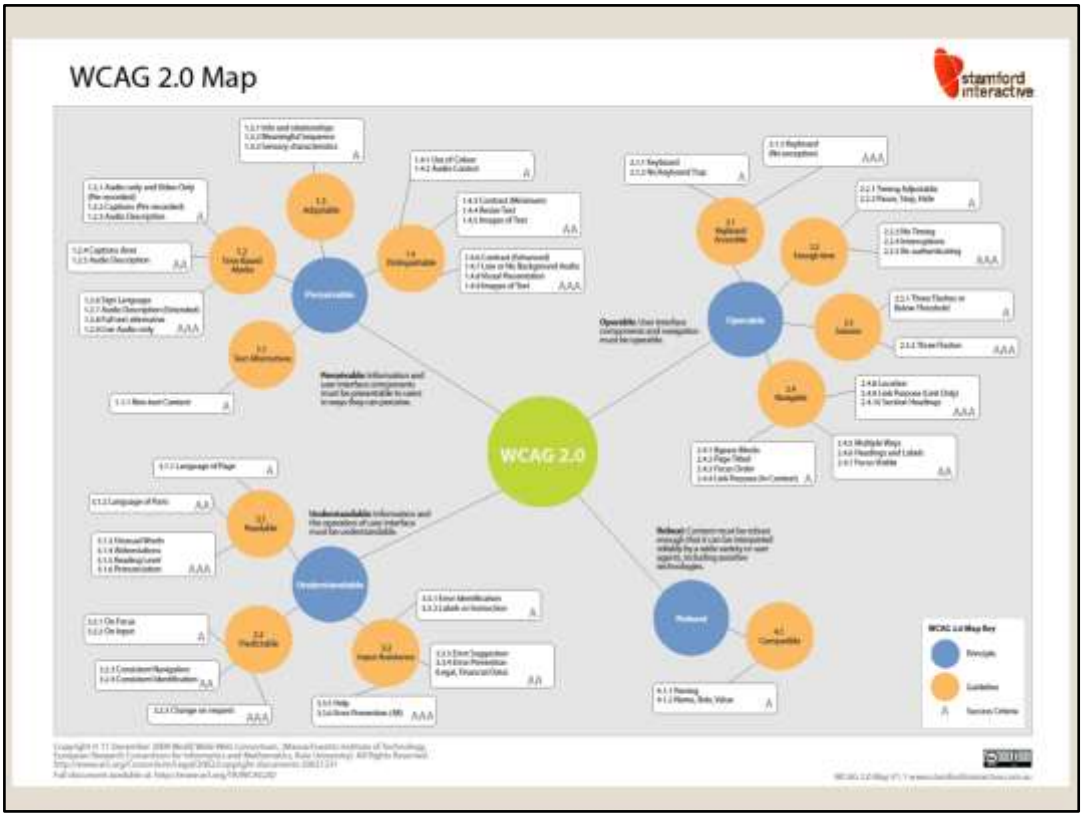
These figures do not include mild conditions such as long sightedness or colour blindness or symptoms experienced by other conditions such as hand tremor.

Web Accessibility in Mind (WebAIM) estimates that 8.5% of people have at least one disability that limits their use of the computer and internet.

Accessibility barriers include people with temporary or permanent impairments including but not limited to:

- Visual impairments – low visual acuity, blindness, low vision, colour blindness, cataracts
- Cognitive impairments – Attention Deficit Disorder, dyslexia, memory loss
- Physical impairments – broken hand, tremors, quadriplegia and repetitive strain injury
- Aural impairments – hearing loss, deafness
- Photosensitive disorders – epilepsy, migraine

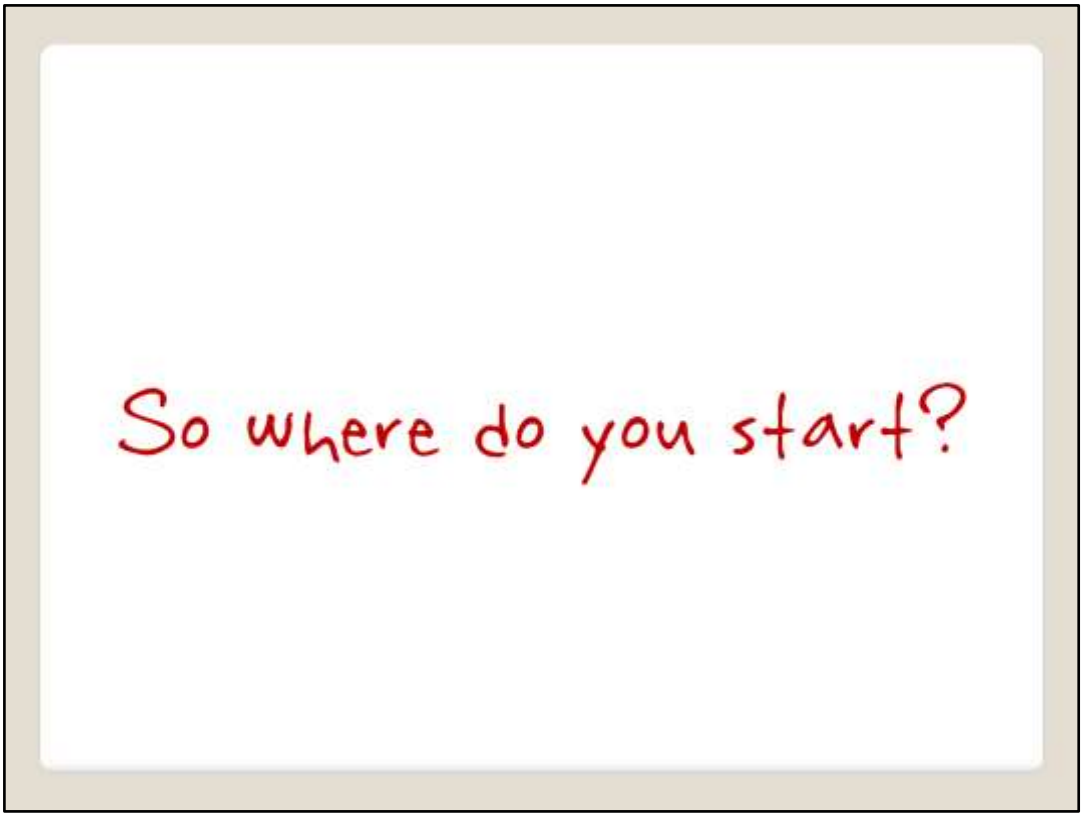
These statistics are consistent with statistics worldwide.



Get a hold of our WCAG 2 map off our website – no it's not a promotional thing, we like it because it actually helps *us* keep track and make sense of WCAG 2, so it could help you too.

There is a deadline for Federal Government in Australia, set out by the National Transition Strategy.

By this time (mid-2011), your department should have already completed its systems stocktake, and developed a strategy towards how to get them compliant.



So where do you start?

As a tester, checking for accessibility can be daunting if you don't know where to start. But you already have most of the tools you need at hand.

Checking for usability after the system is built might make it too late to fix things. But at least you can flag it, and build the culture.

Testing Tip #1: Test with your keyboard.

This in itself will check your users' ability to get to important content in a logical manner. This will affect screen reader and dictation tool users, who will be dependent on keyboard commands for their interaction.

Check for access to:
- shortcuts and bypasses
- correct navigation order
- navigation elements (beware the flyout navs)
- ability to stop and start things
- context help/menu - 'right-click' options
- Buttons
- Links
- Forms

A keyboard user's navigation experience is much more linear than someone skimming with their eyes and a mouse. So check that the keyboard order actually goes in the way the user would expect.

The examples we're about to show aren't all bad – in fact a lot of them have great design aspects and are designed to fit a purpose. But there are a few small things that could always be improved and we just want to show you what to look out for. As you might know by now, it's tricky trying to remember how to test for all angles, especially those you aren't used to.

The FaHCSIA website flow shown here is actually technically correct in terms of its flow despite it looking like it's all over the place. Focus goes to the main navigation options which are on the right – but this is not what the sighted keyboard user expected, as they visually look left-to-right, and so the opposite behaviour "seemed" incorrect.

The St. George loan repayment calculator is a very easy to use and functional widget, but relies on your ability to change focus and manipulate the sliders using your mouse. There has to be a keyboard-friendly version of this.

Testing Tip #2: Turn your settings on or off.

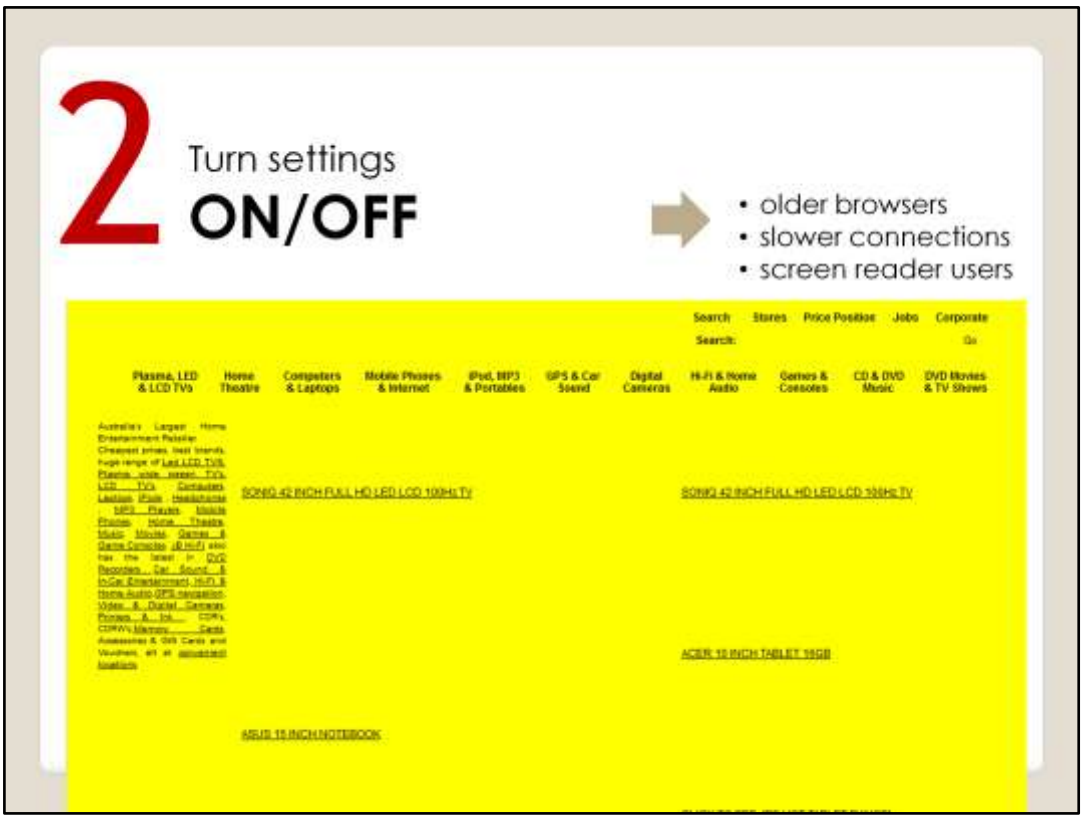
You have a number of settings that you can control on your desktop which can help simulate the experience for users who might not be able to access content types, possibly due to older browsers, or slower connection speeds. This also affects screen reader users.

Try turning off:

- images
- sound
- JavaScript
- Flash

The example shows the JB Hi Fi website with its colourful splashes of electronics and sales.

But if you turn off images in your browser...



The JI Hi Fi website with images turned off simply gives you a list of links.

What information was lost? Oh yeah – the sale prices. They were embedded in the images.

Captchas can be an “accessibility fail”. Turn images off, and the test disappears, as to do the icon options for alternative options.

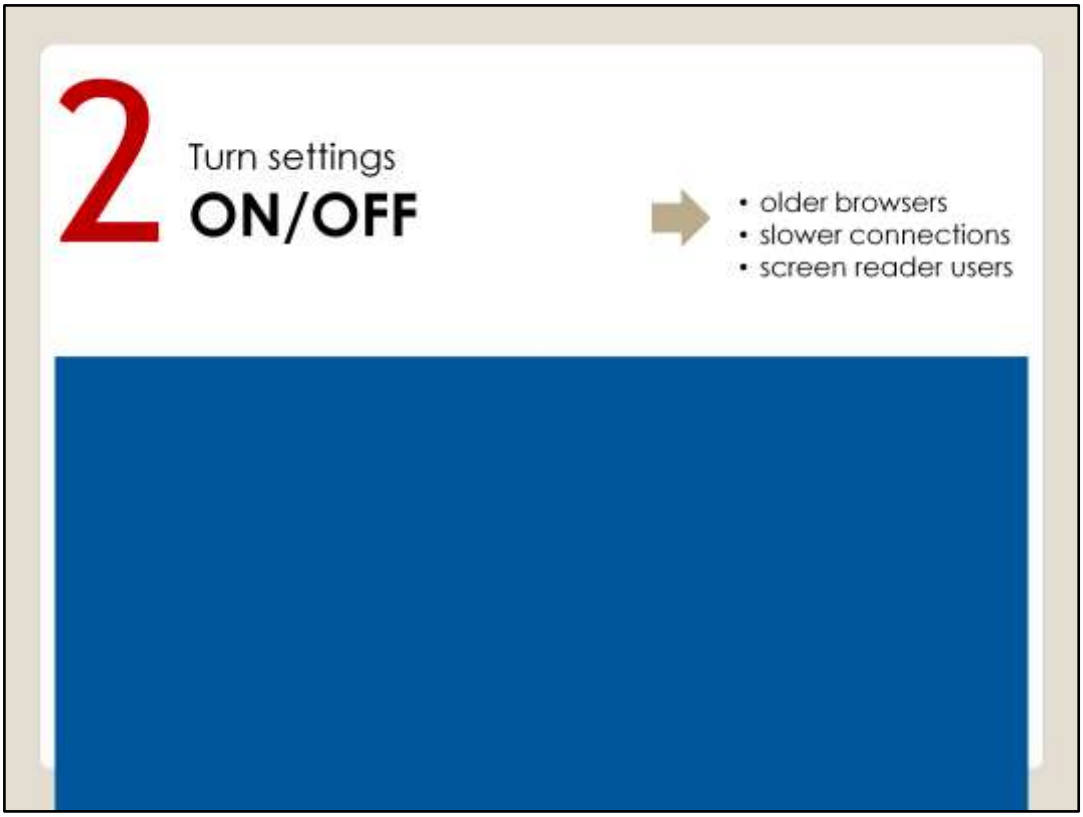
This immediately stops someone from being able to progress with interacting with the site.

On the left is an example video from “A Current Affair”. No captions or text transcript are available. The assumption is that you can listen to the audio. But how often have you chosen *not* to, because it would disturb others around you, or you forgot your headphones, or your speakers don’t work? There is no other way to find out what is happening in the story.

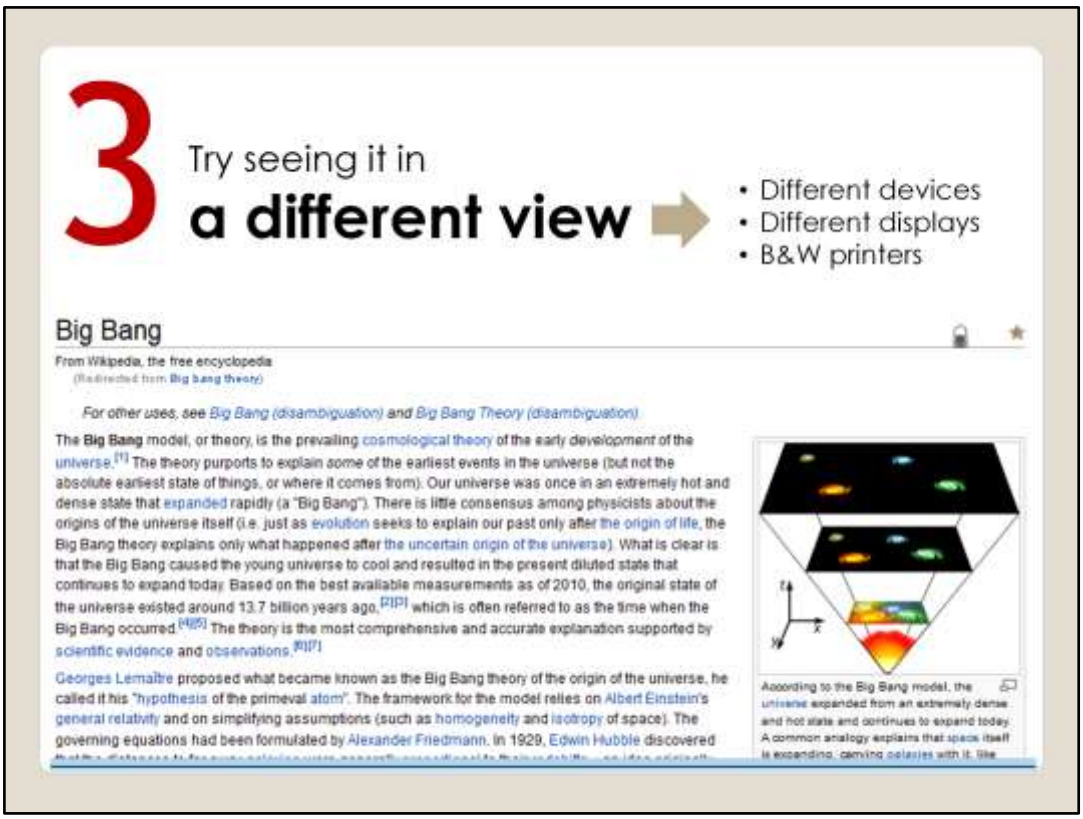
On the right is a YouTube clip from the Equals3 channel. Important elements are highlighted within the clip, and YouTube provides a captioning function that you can correct and improve.

You can have an incredibly fancy site running on Flash, which is useful because you can design it a specific way and have it render the same in any browser. This Disney site is highly interactive.

But disable Flash, and you get...



A blank screen. No hints for what to do next. No access.

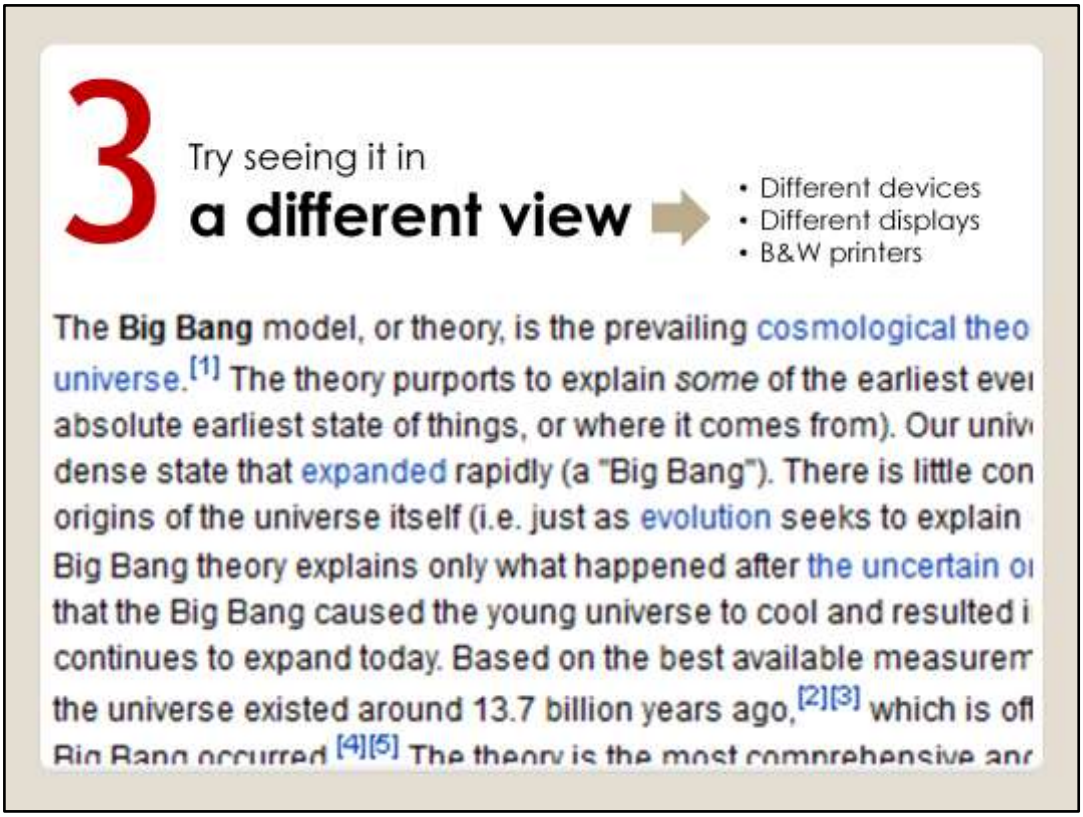


Testing Tip #3: Try seeing it in a different view.

Users have different display requirements, depending on their devices or preferences. A standard webpage such as this wikipedia page may be what most users see, but that is not the only way it may display.

Try:

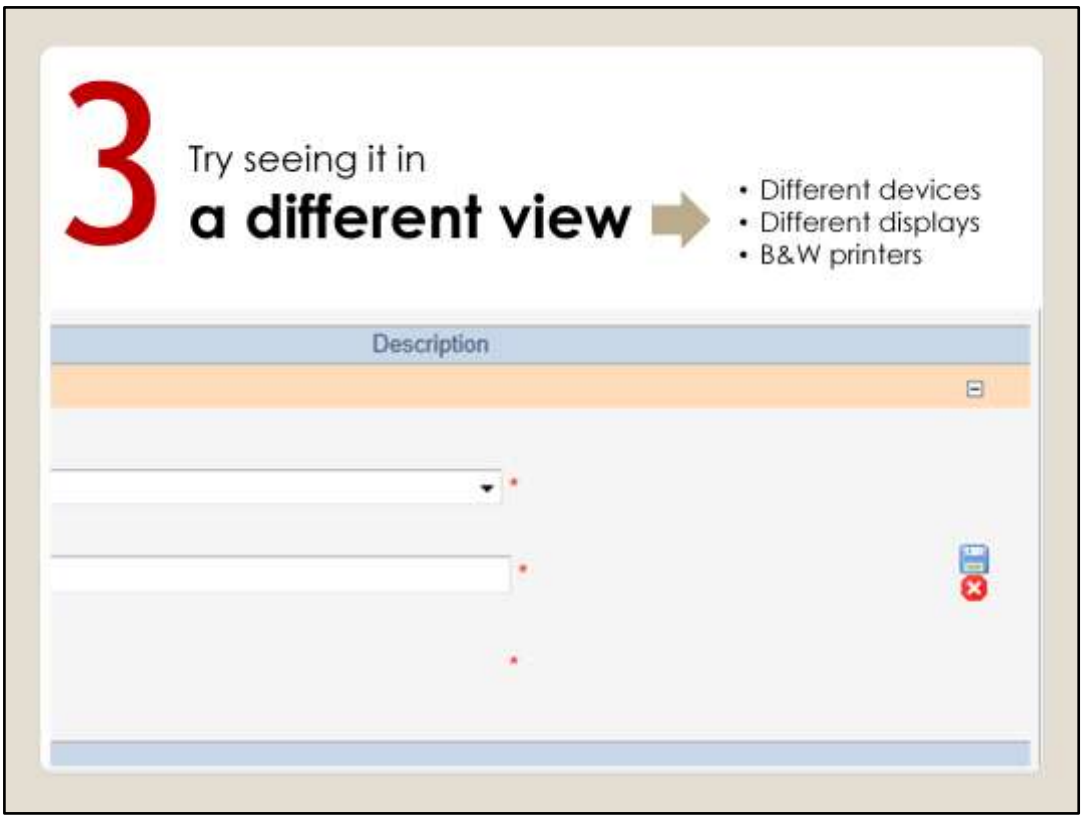
- different browser
- larger resolution
- smaller resolution
- mobile display
- high contrast



Users might need a larger display to help readability, so may opt for magnification. What does this do to the rest of the page?

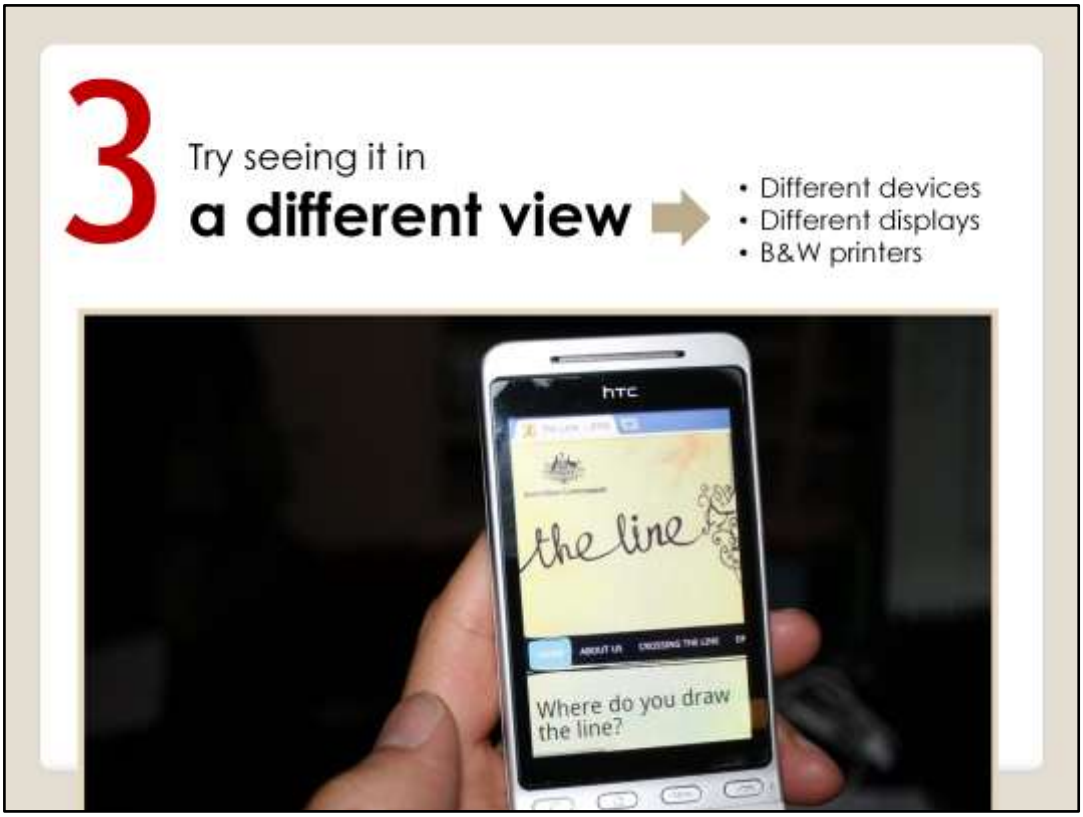
When content is magnified, the user loses the bigger picture of what else is available on screen.

We are seeing more devices adopt smaller screen sizes, as we have a large aging population whose eyesight will deteriorate over time.

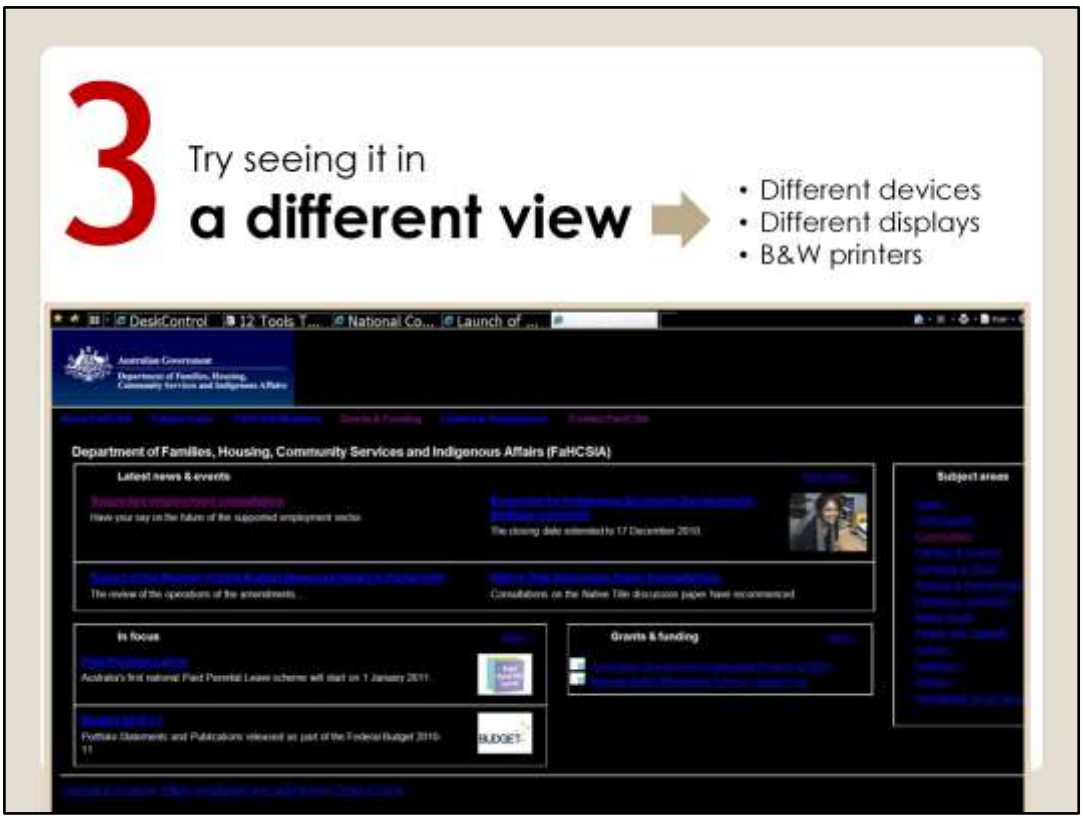


Content that is highly magnified becomes hard to use if the user isn't given clear context through good design.

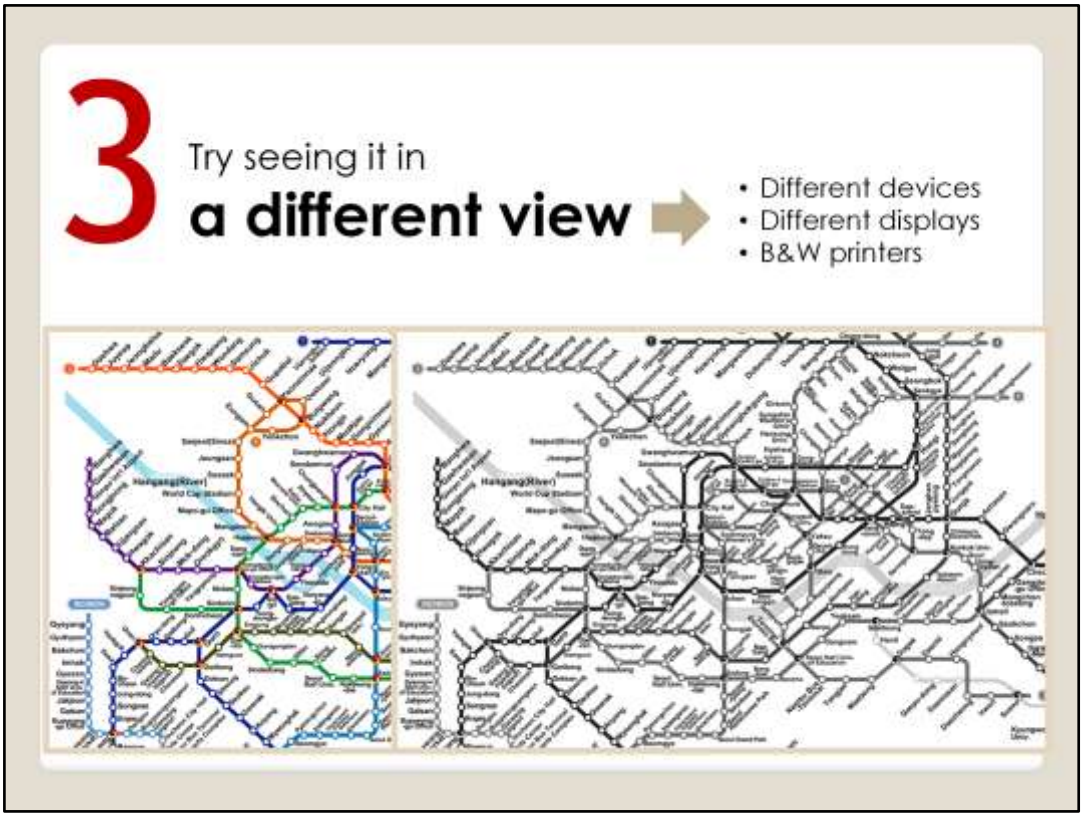
This image shows a form with the text labels cut off the screen, but the mandatory indicators are visible to the right of the labels. This means the user has to scroll to the right to see which are mandatory, and scroll back left to see what the field label is. Both pieces of information are not visible at the same time, making context difficult.



You don't necessarily know how a user is accessing the content. If the site or service can be accessed via a mobile device, try it out and see what the experience is like.

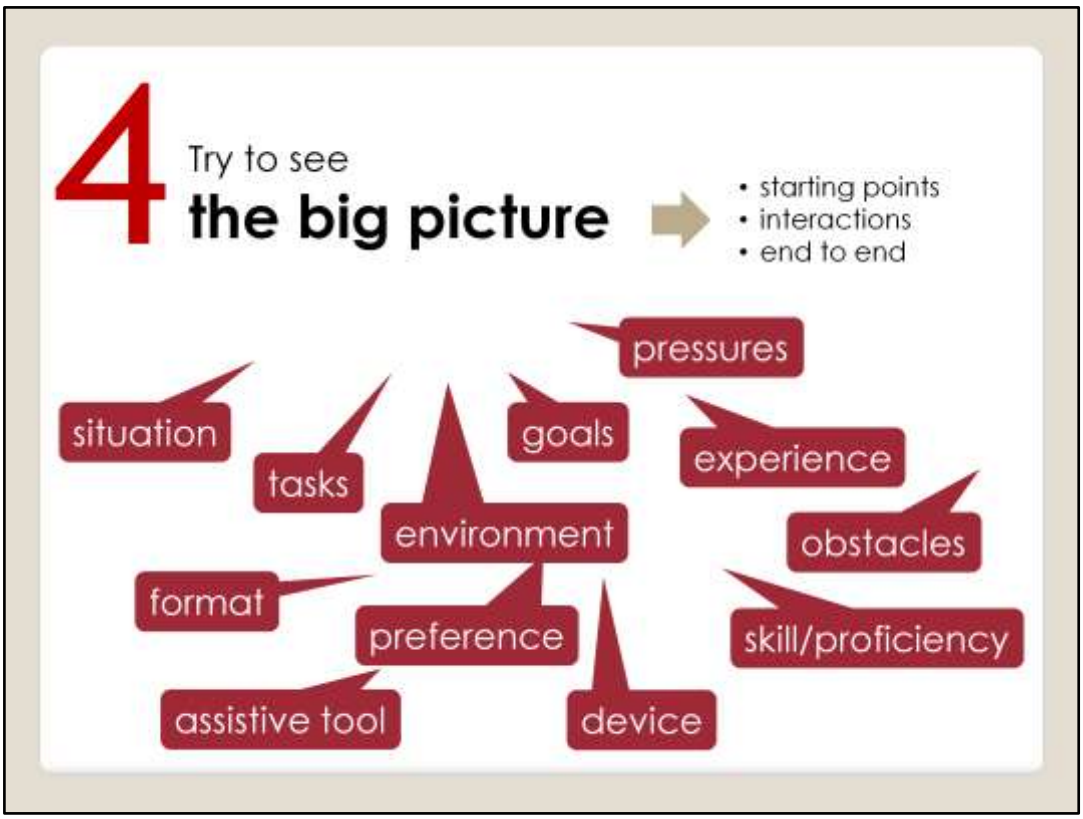


Try a high contrast view to see what content does and doesn't transform. Someone who uses high contrast views may have their views set up a specific way so what you can check for is that everything that is important does convert.



Information that is represented by colour might get lost if you print to the black-and-white printer, or if the user has a deficiency in their colour perception.

The metro map on the left is highly colour-dependent, and could have become unusable in its grey-scale version on the right. But because the greyscale has enough contrast to be distinct, and the lines and colours are coupled with distinct labels, the information is meaningful even in the absence of colour.



Testing Tip #4: Try to see the big picture.

This is where it's important to remember what the user needs to be able to do. Remember how we said "it depends"?

Is there consistency across the site/application/product?

What are the situations in which a user might be interacting with it?

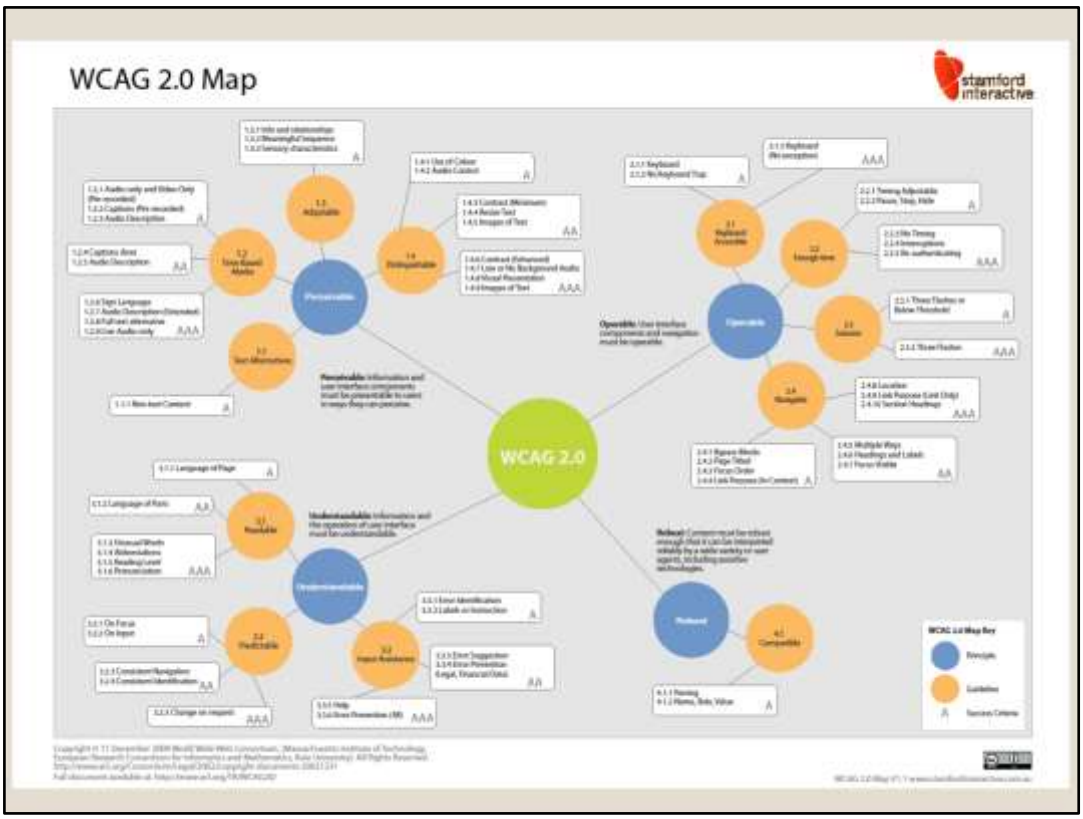
If one piece doesn't work, what is the equivalent or alternative way to perform the same task? What other formats do they get?
What is plan B? And does it work well?

This is where the bulk of work lies.

Accessibility needs to become part of the regular QA process for software testing. We were meant to do this years ago, and we're only just catching up now. This needs to be part and parcel of testing.

It's also important to remember that usability is critical. A function might "pass" accessibility criteria, but the user experience might be awful. What we always end up doing is reporting on both. Something might be accessible, but we also add our recommendations of how it could be more usable. We encourage you to do the same. Even if it lies outside the scope of the test – know that if something is annoying *you* as a tester, its effect on the end user is bound to be bad as well. Take the opportunity to report any of these, and start to build a culture of quality, where BAs and developers can expect you to actively provide feedback on design and user experience, not just on functional passes or fails.

And last but not least, there are legal repercussions if we don't get this right. Complaints may not just come from irate users from the general public complaining about your website or the quality of your tool. They may very well (and do) originate from your own staff and fellow employees in your own organisation, if they find themselves saddled with software they are unable to effectively use to do their jobs, and it affects their performance.



Just a reminder – if you don't have this already, grab the WCAG 2 map off our website.

Be careful. You can do your research, know what your users want, design the best, most usable “thing”, have the best intentions – aah, but can they get to it?

(You gotta love failblog.org)

We hope this helps. It's only an introduction into the type of work that needs to be done, but we hope it's enough to get you started.

If you have any questions or want further information about any of this, we're always delighted to help!