



BBC iPlayer Web's approach to accessibility



Andy Smith
Software Engineering Team Lead
@AndyMSUK

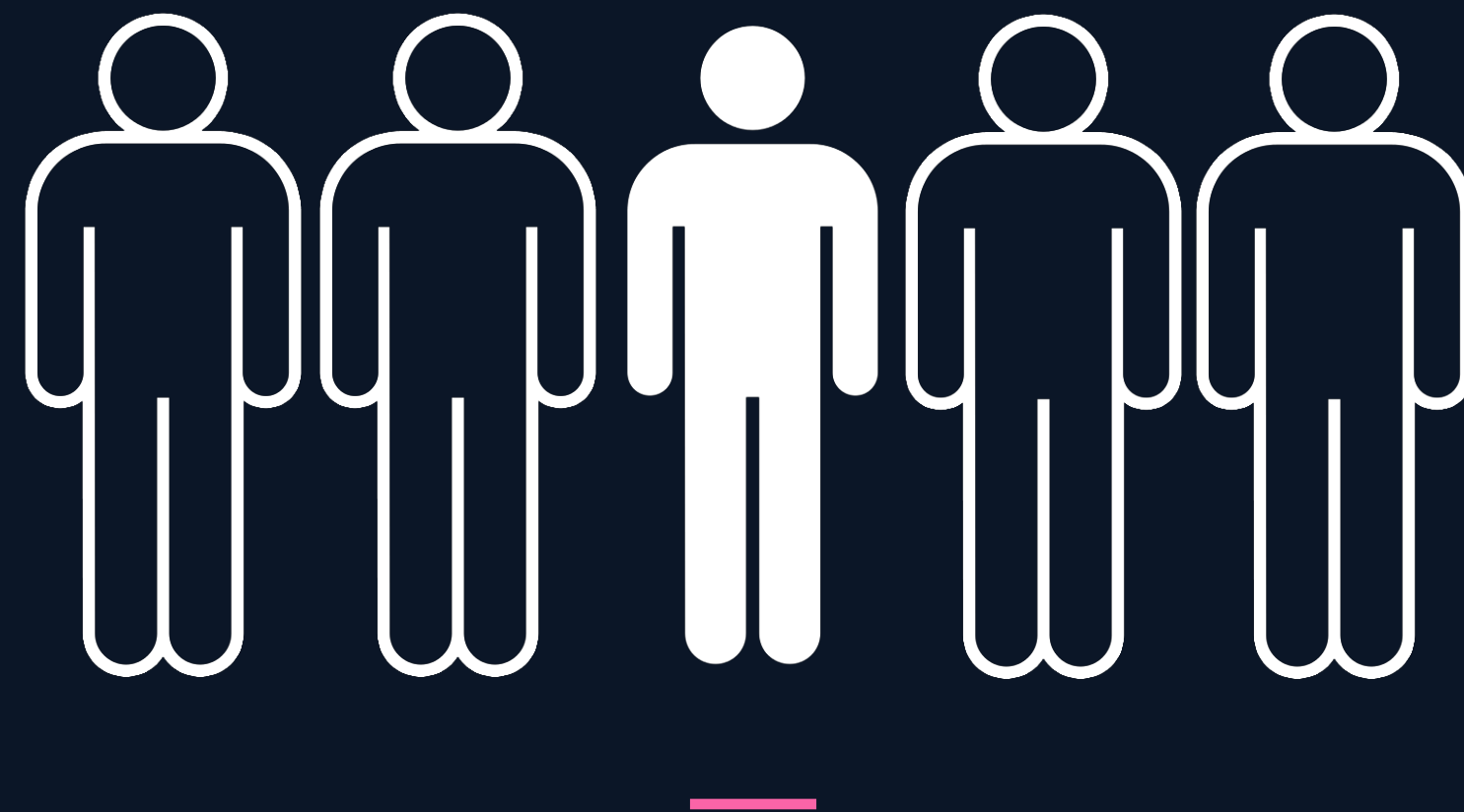


Emily Atkins
Senior UX Designer
@Em_Atkins

Long-term / permanent disability



At least 1 in 5 people in the UK have a long term illness, impairment or disability



Source: Department for Work & Pensions

Temporary disability



Situational disability



Everyone has accessibility needs



**“The power of the Web is in
its universality.
Access by everyone
regardless of disability is an
essential aspect”**

Tim Berners-Lee - Inventor of the World Wide Web

40% of local council sites failed basic accessibility tests

70% of UK websites
are inaccessible

Source: Business Disability Forum

New regulations came into force for public sector bodies in 2018 that say you must make your website or mobile app accessible by making it ‘perceivable, operable, understandable and robust’

Coming Up

1. Designing for Accessibility
2. When to think about accessibility
3. Development processes
4. Automated Testing
5. Manual Testing
6. Team Culture
7. Q&A

Designing for Accessibility

Designing for users who are deaf or hard of hearing



Do...

Don't...

write in plain language

Do this

use complicated words or figures of speech



use subtitles or provide transcripts for videos



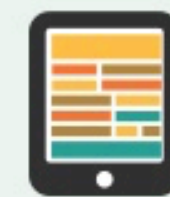
put content in audio or video only



use a linear, logical layout



make complex layouts and menus



break up content with sub-headings, images and videos



make users read long blocks of content



let users ask for their preferred communication support when booking appointments



make telephone the only means of contact for users



Designing for users on the autistic spectrum



Do...

Don't...

use simple colours



use bright contrasting colours



write in plain language

Do this

use figures of speech and idioms



use simple sentences and bullets



create a wall of text



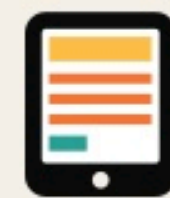
make buttons descriptive

Attach files

make buttons vague and unpredictable

Click here!

build simple and consistent layouts



build complex and cluttered layouts



Designing for users with low vision



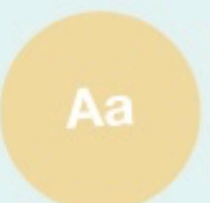
Do...

Don't...

use good colour contrasts and a readable font size



use low colour contrasts and small font size



publish all information on web pages



bury information in downloads



use a combination of colour, shapes and text

Start

only use colour to convey meaning



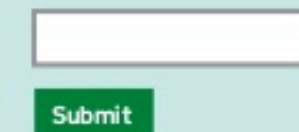
follow a linear, logical layout



spread content all over a page



put buttons and notifications in context



separate actions from their context



Designing for users with dyslexia



Do...

use images and diagrams to support text



align text to the left and keep a consistent layout



consider producing materials in other formats (for example, audio or video)



keep content short, clear and simple



let users change the contrast between background and text



Don't...

use large blocks of heavy text



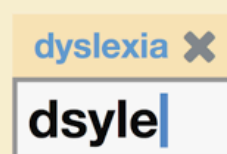
underline words, use italics or write in capitals



force users to remember things from previous pages - give reminders and prompts



rely on accurate spelling - use autocorrect or provide suggestions



put too much information in one place



Designing for users with physical or motor disabilities

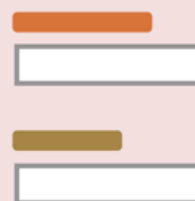


Do...

make large clickable actions



give form fields space



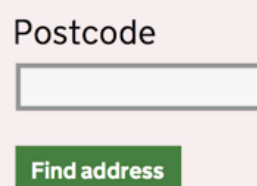
design for keyboard or speech only use



design with mobile and touchscreen in mind



provide shortcuts



Don't...

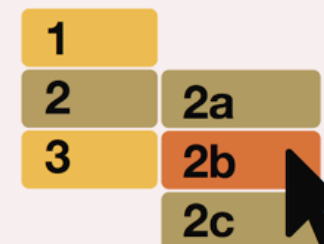
demand precision



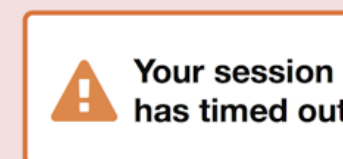
bunch interactions together



make dynamic content that requires a lot of mouse movement



have short time out windows



tire users with lots of typing and scrolling



Designing for users of screen readers



Do...

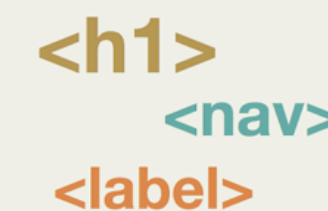
describe images and provide transcripts for video



follow a linear, logical layout



structure content using HTML5



build for keyboard use only



write descriptive links and headings



Don't...

only show information in an image or video



spread content all over a page



rely on text size and placement for structure



force mouse or screen use



write uninformative links and headings



Logical reading order

Keep the reading order logical, to enable screen readers to read the text aloud and in order.

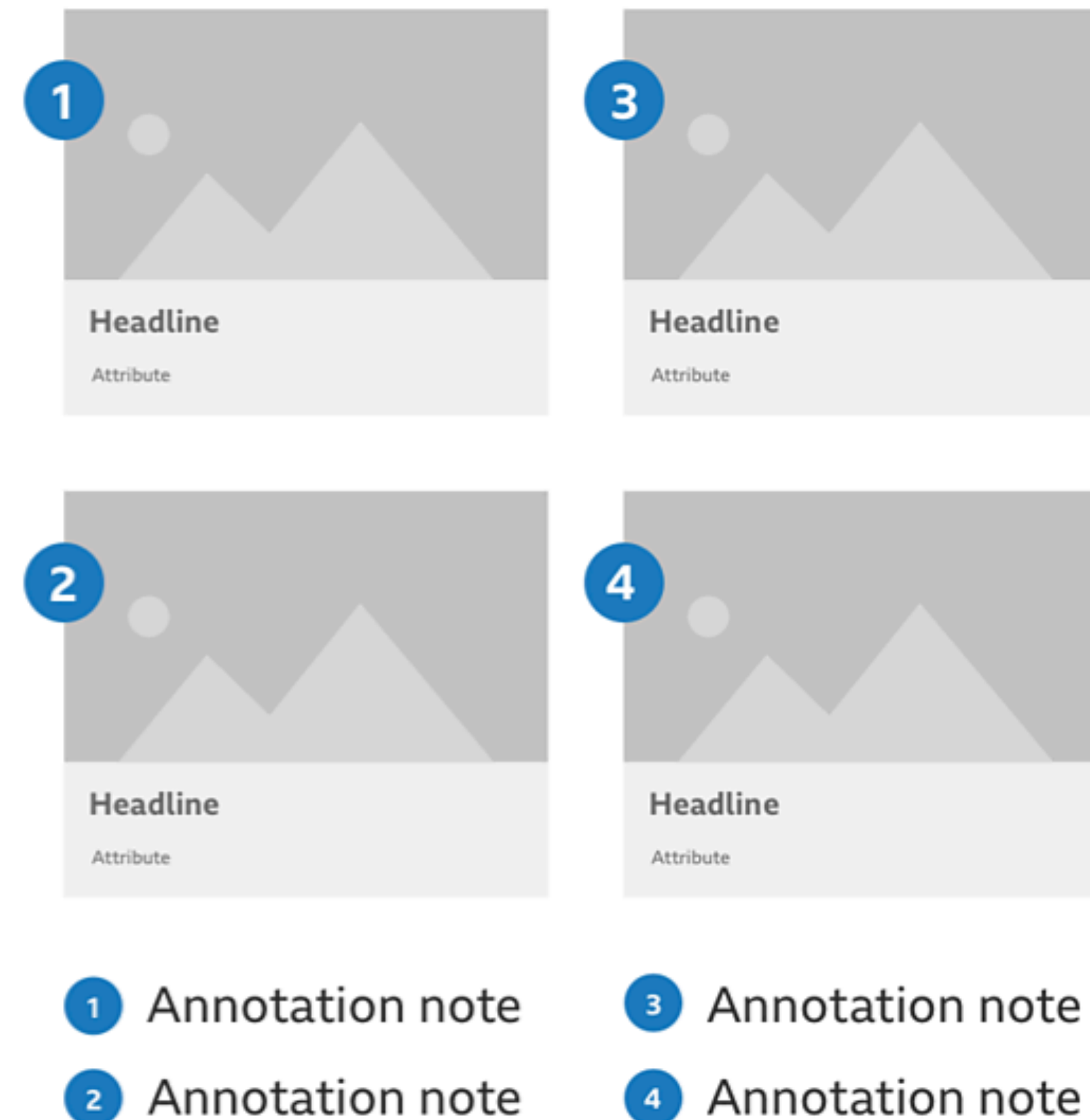
Heading tags

When headers are not present the sense of structure is lost and screen readers have no headers to read out to the user.

Meaningful link text

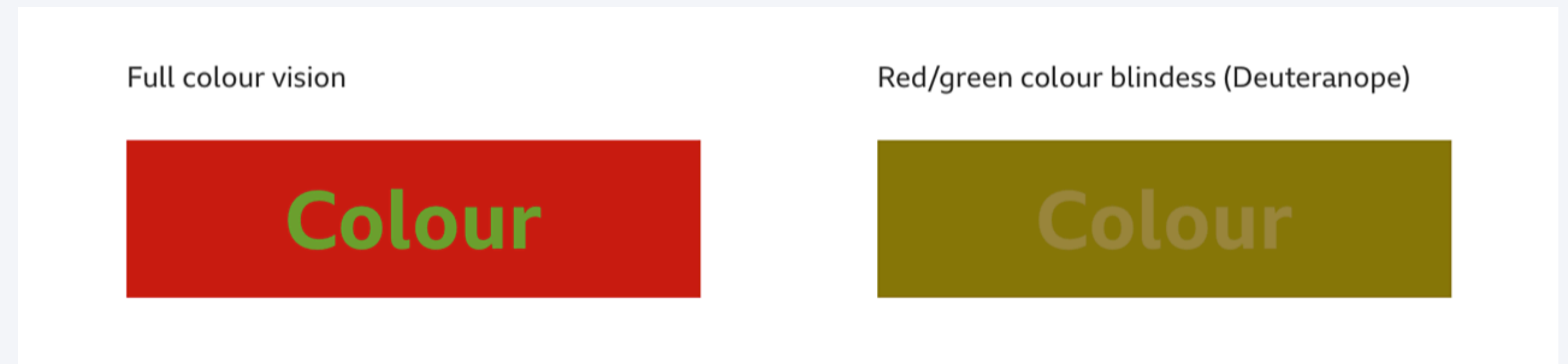
Avoid generic link text such as 'click here' and 'more'.

Number annotation can show tab order and include additional notes



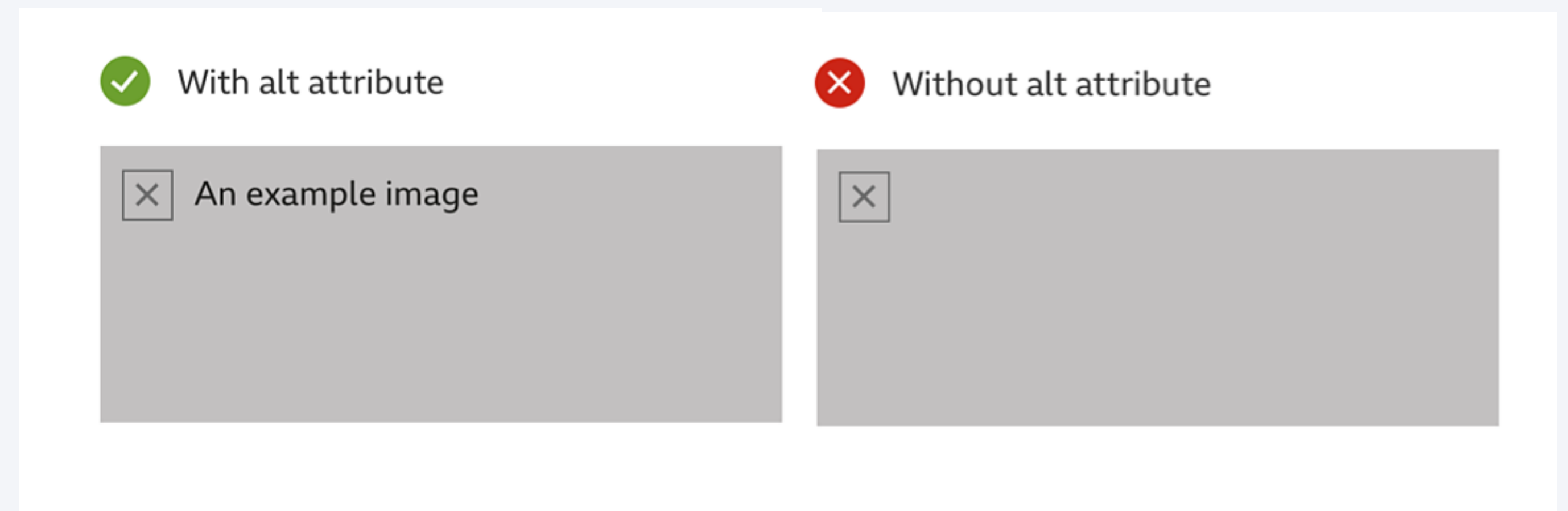
Colour

Avoid having the text and background in similar colours.



Alternative text for images

Add alternative text (or alt attributes) to imagery to enable screen reader users to gain the full experience.



Font type and size

Consider the font size for visually impaired users. Anything below 14pts in a browser can be difficult to read.

GEL Guidelines

The screenshot shows the BBC iPlayer website's GEL (Global Experience Language) Guidelines section. The navigation bar includes 'Home', 'News', 'Sport', 'Weather', 'iPlayer', 'Sounds', 'CBBC', and 'More'. The main navigation includes 'Home', 'Guidelines', 'Articles', and 'About UX&D'. The sub-navigation includes 'Foundations', 'Design Patterns', and 'How-tos'. The article title is 'How to design accessible email newsletters'. The sub-header is 'How to use accessibility principles to design email newsletters that everyone can use.' The contributors are listed as three individuals and '+1'. The article was published on 26 Jan 2017.

Contents

Why design for accessibility?

Getting started

Key considerations

Annotating wireframes and UX designs

User testing

Further information

Getting started

If accessibility is not something you've had to consciously think about before, here are a few things to keep in mind.

Put people first

People are different. So are impairments. They may be cognitive, motor, hearing or vision related, or a combination of these. They may be temporary or long-term, mild or severe. And someone's experience with assistive technology can vary equally as much.

Aim for no impassable barriers to someone interacting with what we build.



Use familiarity

Where possible, use standard user interface elements and familiar design patterns, interactions, editorial, text equivalents and icons. This should be done across apps and sites, so that content and interactions quickly become familiar and recognisably BBC.

For example, consistent labels and text equivalents used with an icon or logo across platforms will help people navigate confidently and reinforce meaning and brand.

Give control

Always give people control over content. It can help them understand what they are seeing and stops them becoming disoriented or lost. This also means not suppressing device features like

How to design accessible email newsletters

How to use accessibility principles to design email newsletters that everyone can use.

Contributors



Published on 26 Jan 2017

Why design accessible email newsletters?

il

Everyone is different and there are many variations of impairment. At the BBC we need to make our content accessible to all, whether they are internal or external users.

In the UK, around 18% of the population are impaired in some way. Approximately 1/3 are temporarily impaired due to illness, injury or situation. And situational impairments can affect anyone at any time. For example someone in bright sunlight may have the same requirement as someone who experiences visual disturbances with migraines or someone who is colour blind.

In order to make your newsletter content accessible to all, you need to consider a wide range

New font

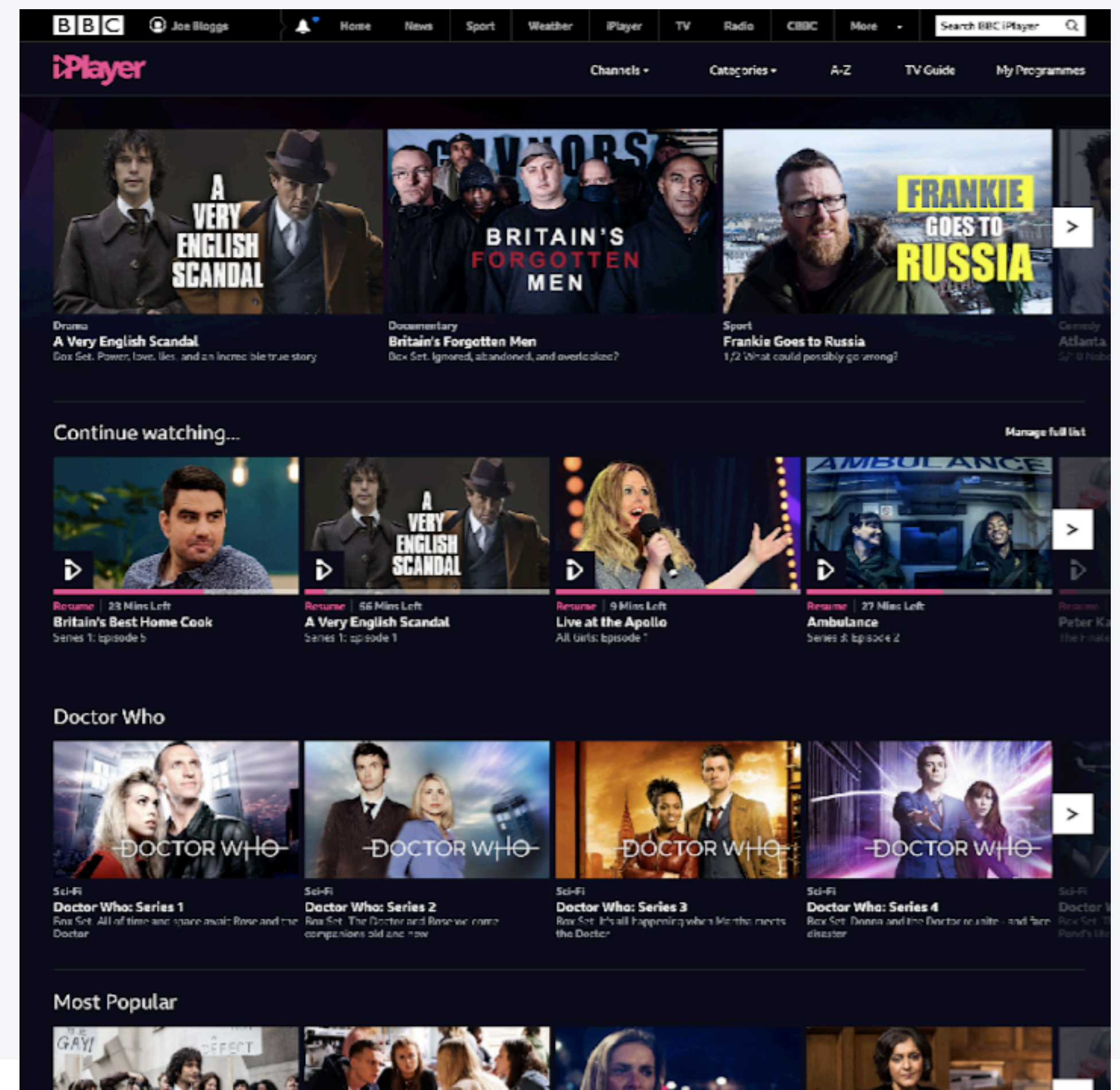
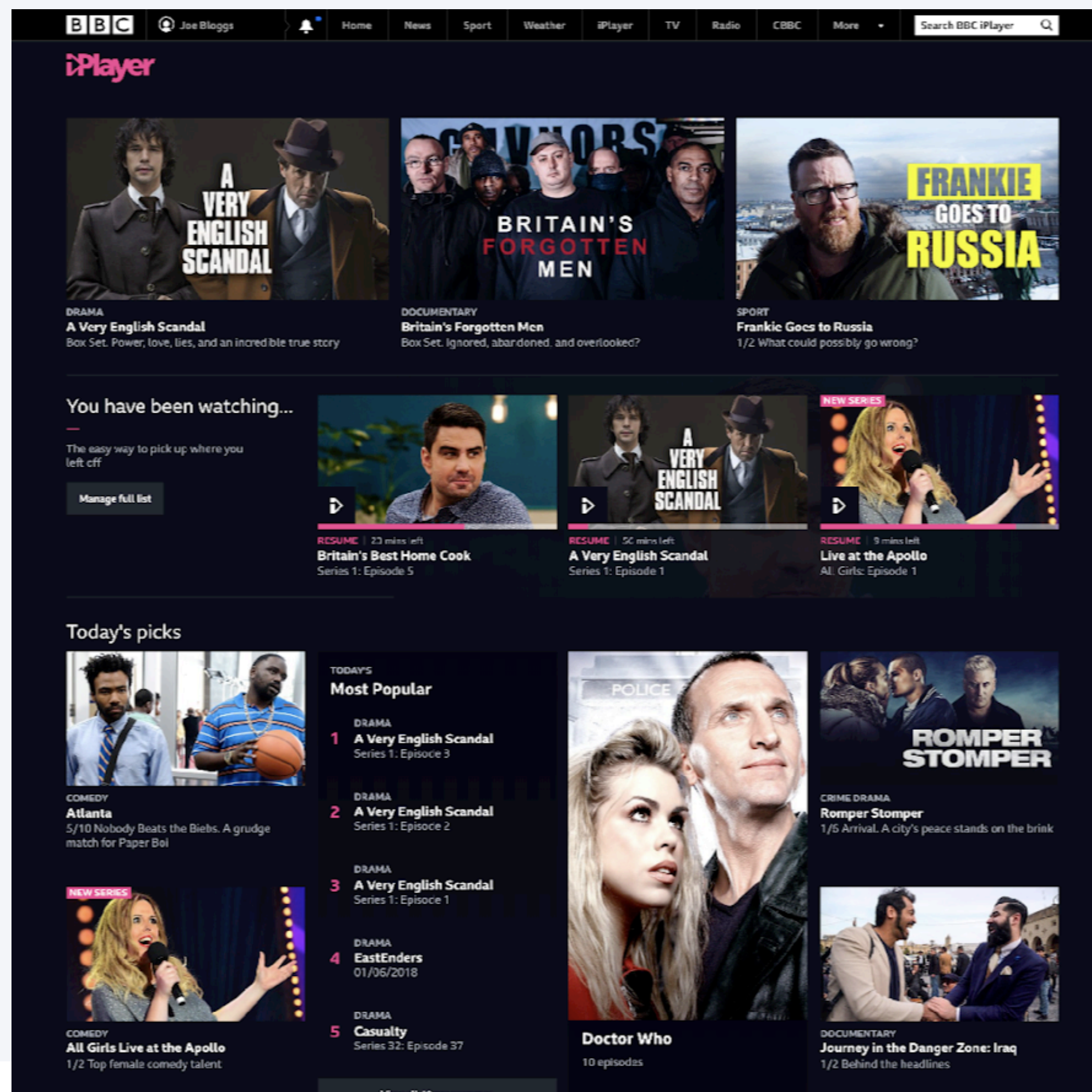
We are now using the BBC's new font - BBC Reith

- Old fonts:
 - Work well in print, but not always clear enough when they appear in small, digital, spaces.
- BBC Reith:
 - Screen-first font, which is clearer and easier to read, especially on smaller devices.

Light
Regular
Medium
Bold
Extra Bold

More simple consistent layout

We're also working to make the layout of the site more consistent and simple

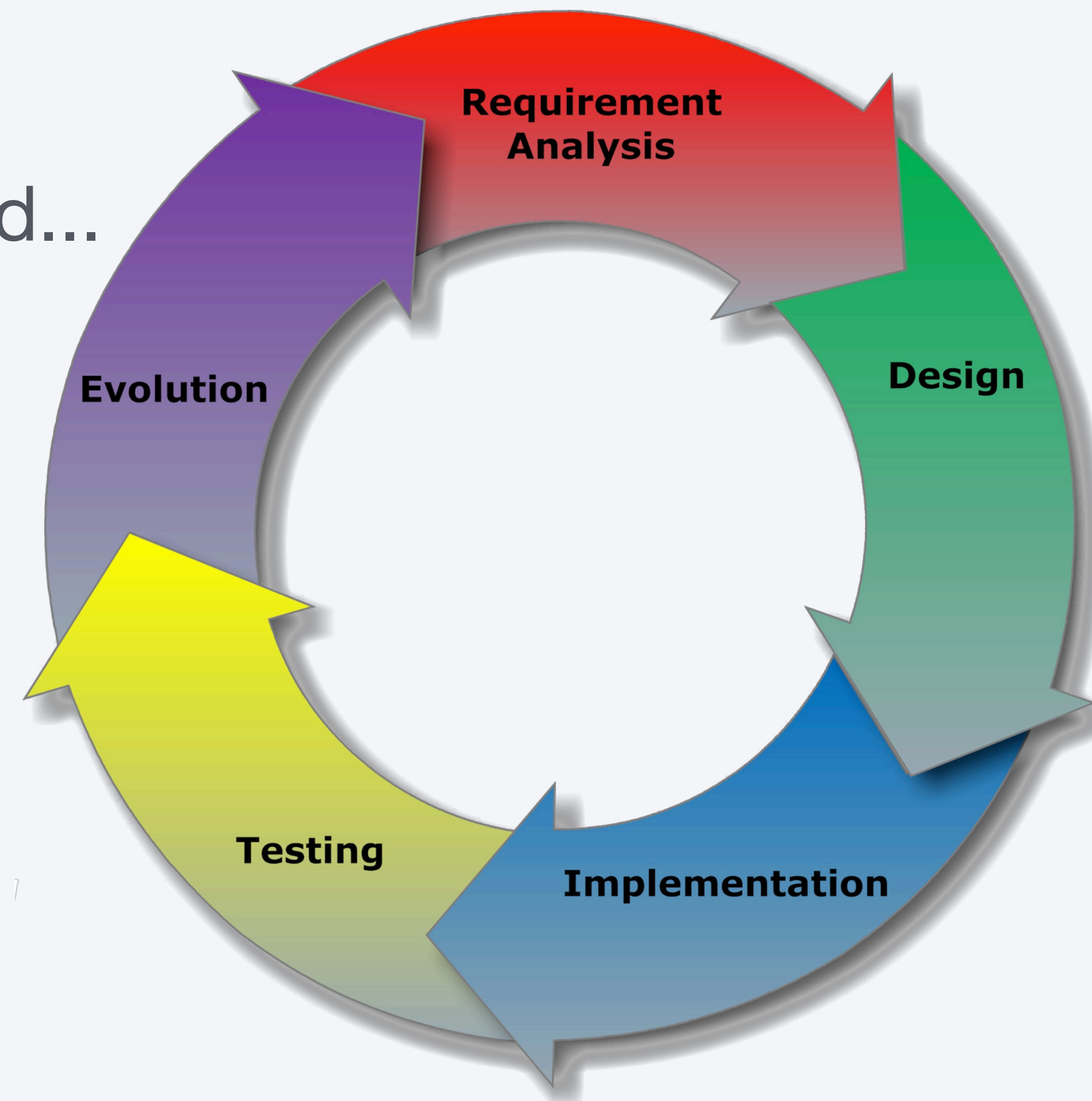


When to think about accessibility

When to think about accessibility

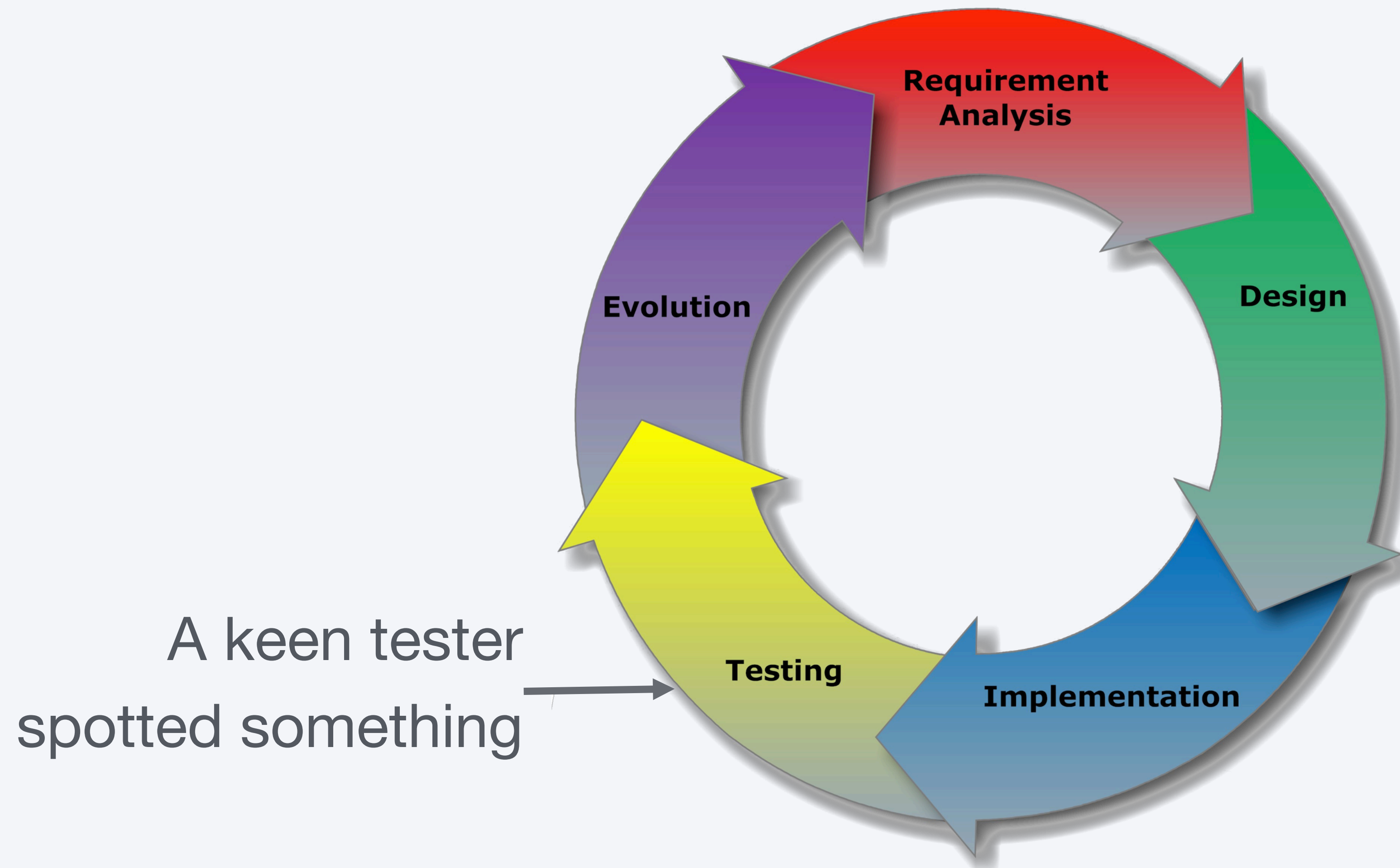
Where do you think about it?

People complained...



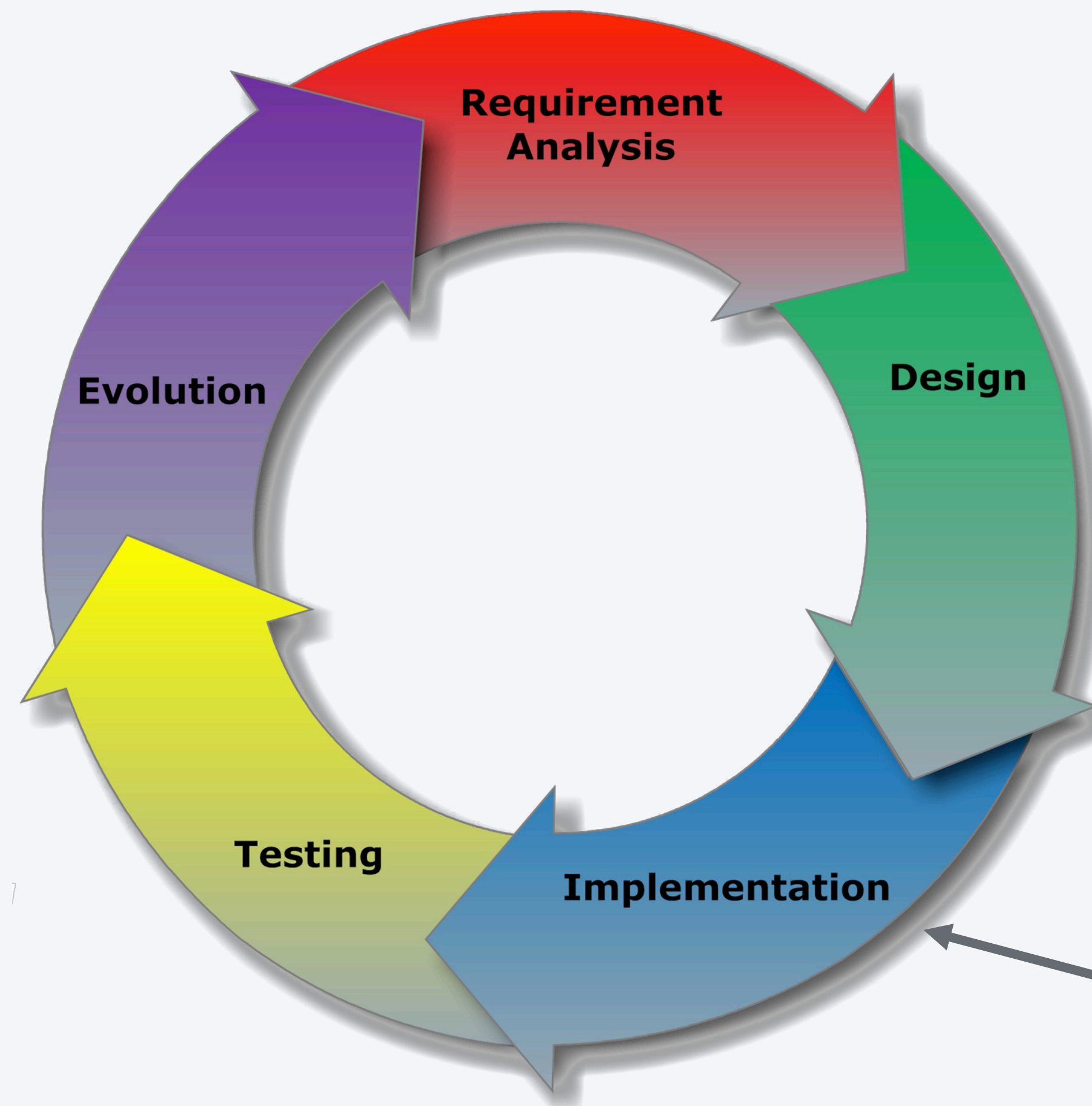
When to think about accessibility

Where do you think about it?



When to think about accessibility

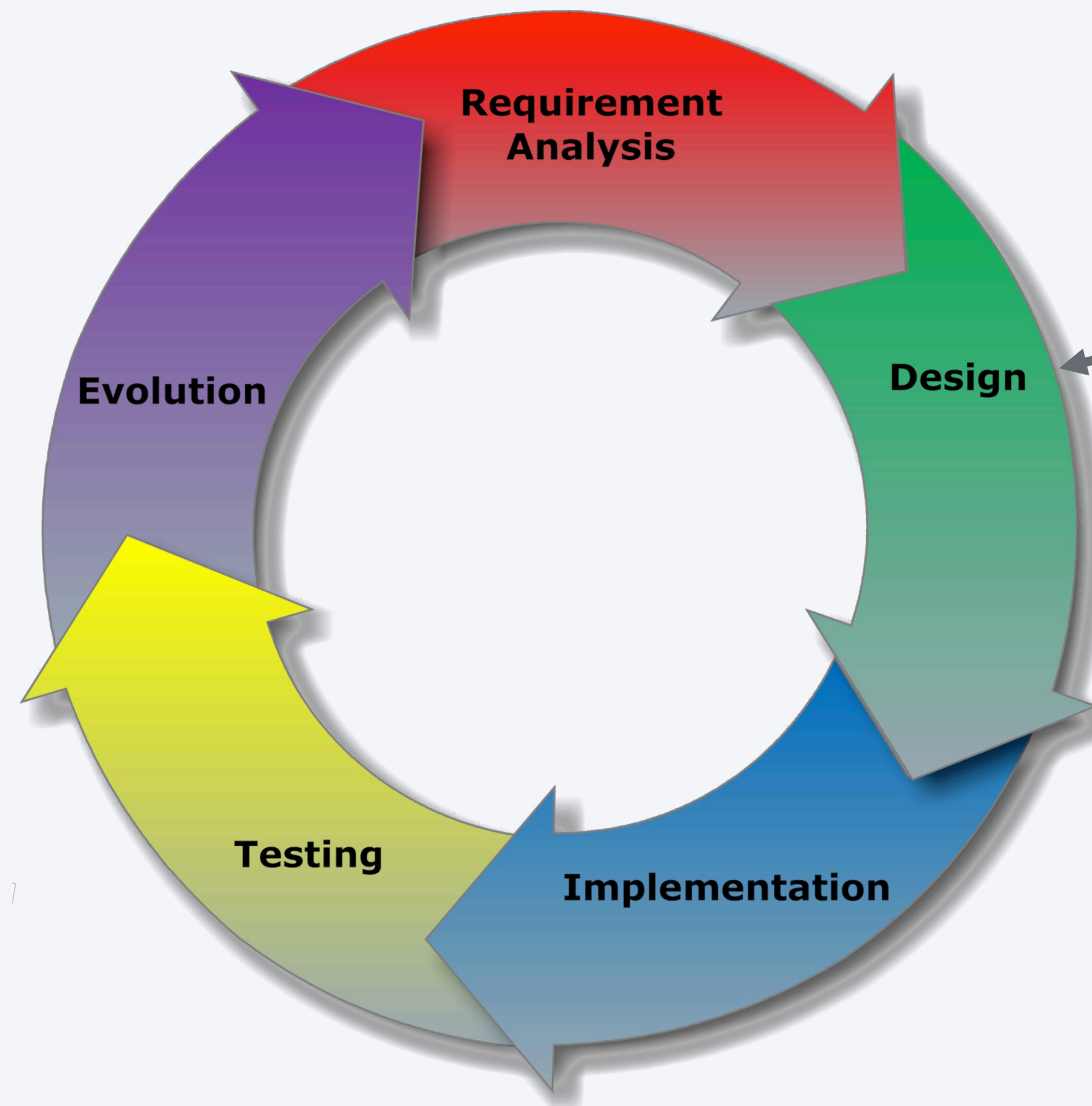
Where do you think about it?



A keen developer
that knows a bit
about accessibility

When to think about accessibility

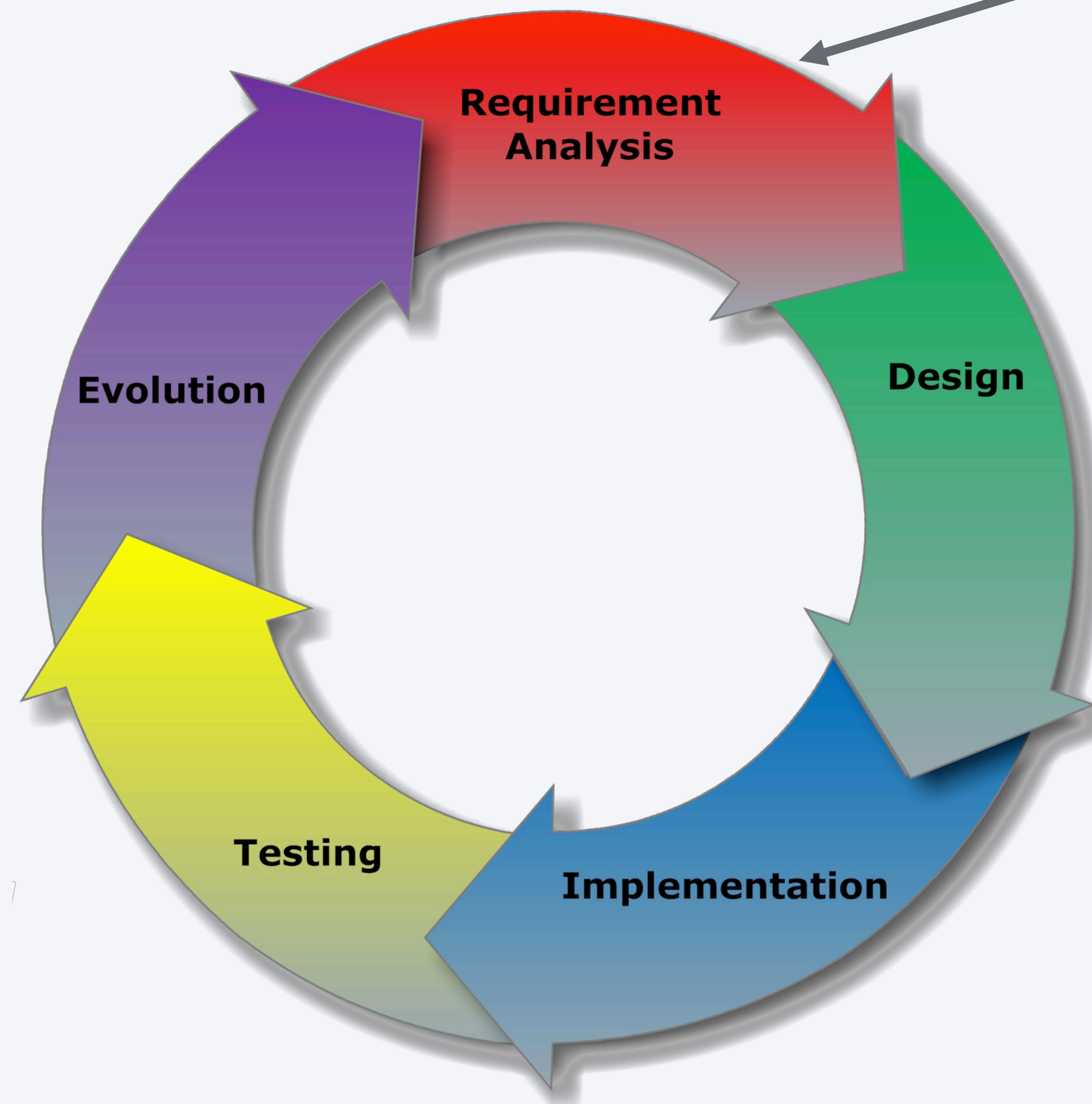
Where do you think about it?



A keen UX designer that knows a bit about accessibility

When to think about accessibility

Where do you think about it?



A keen product owner that knows a bit about accessibility

When to think about accessibility

Where we think about it!

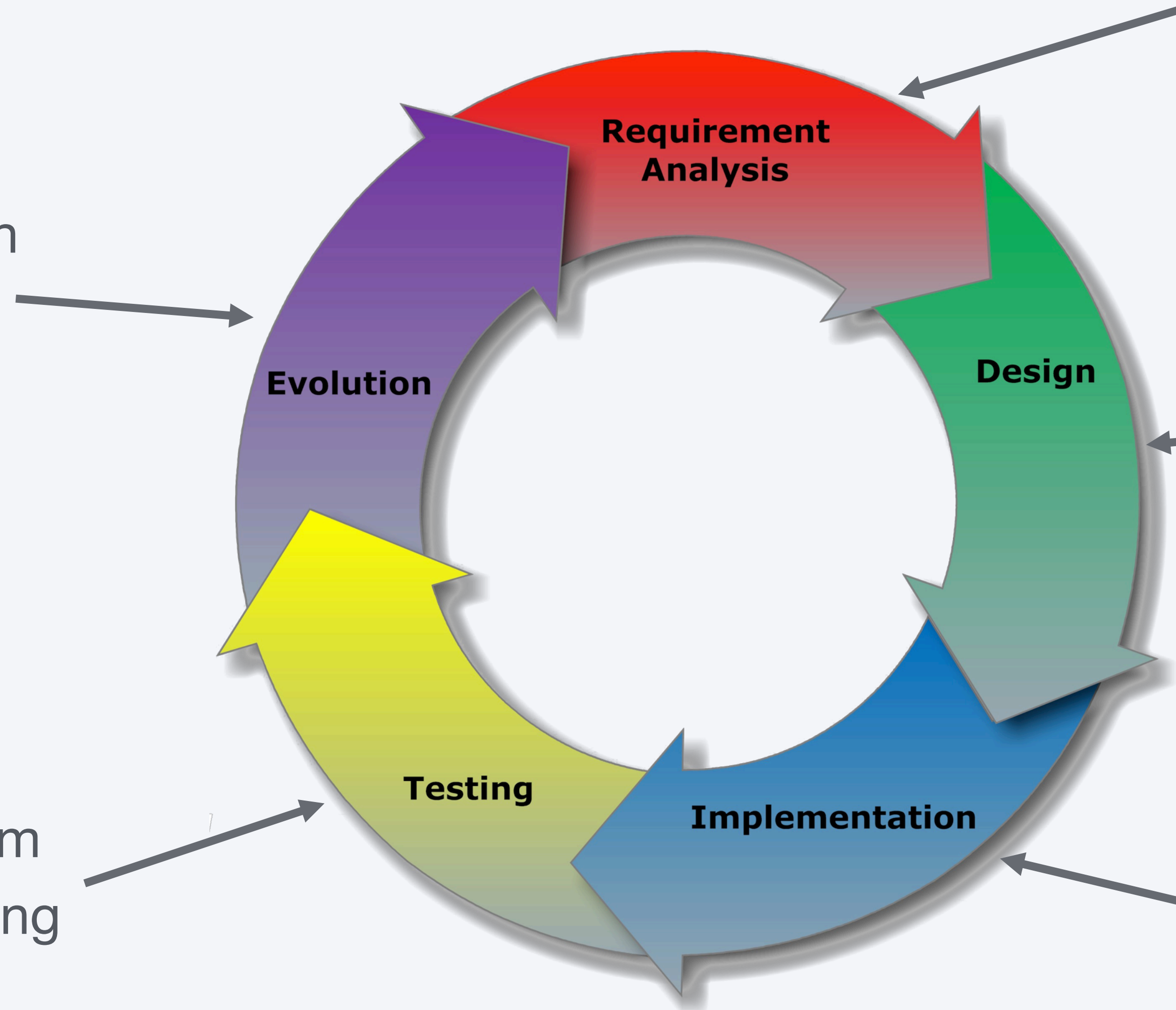
Whole team when discussing requirements

Whole team when maintaining

Whole team when designing

Whole team when testing

Whole team when developing



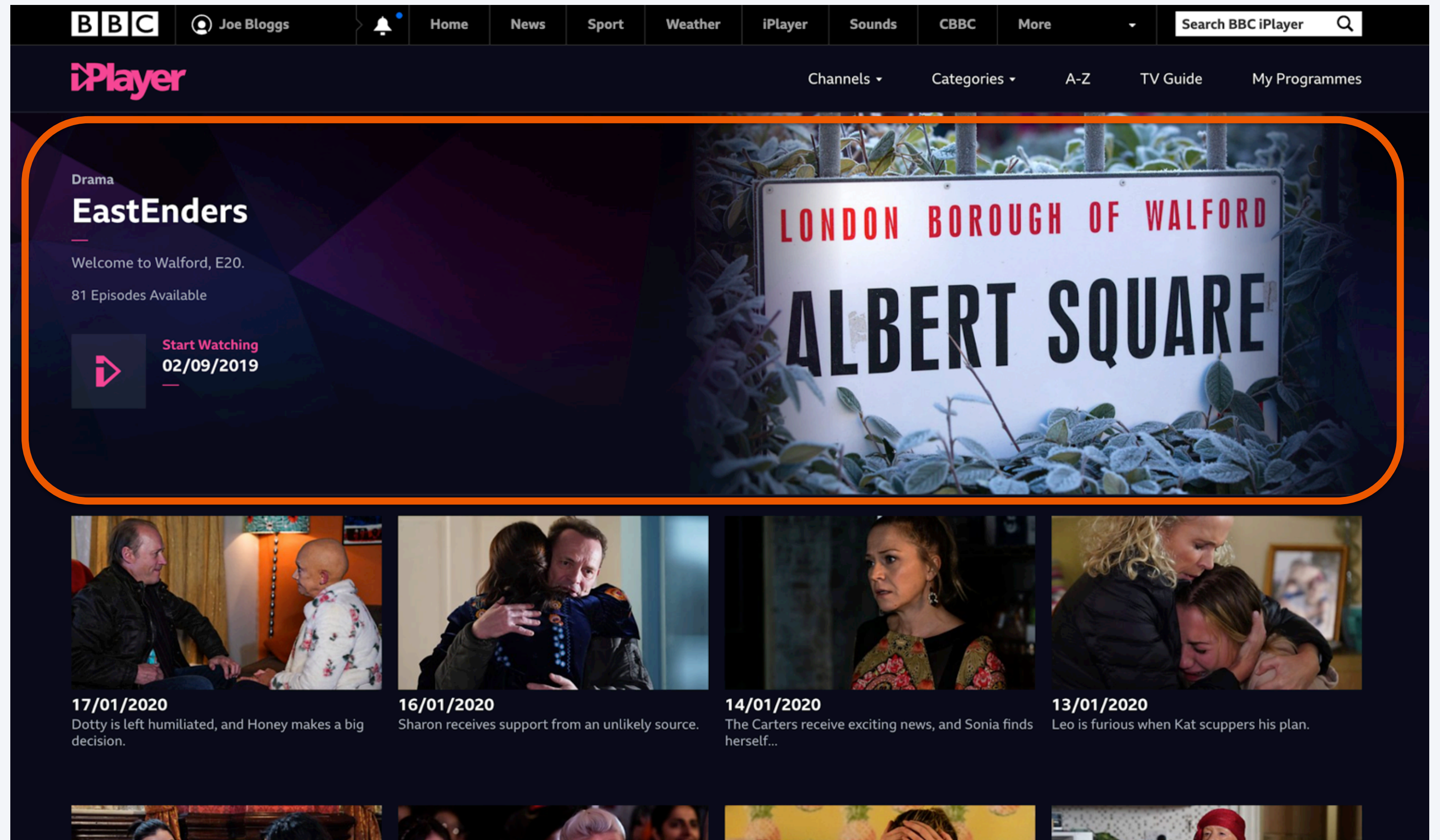
Development Processes

Components

We build documented components so that our developers can write simple code to create accessible web pages

Hero Header

- Our new page header component
- Sets the “title” as an `<h1>` so when this component is used, there’s always a main heading in the markup

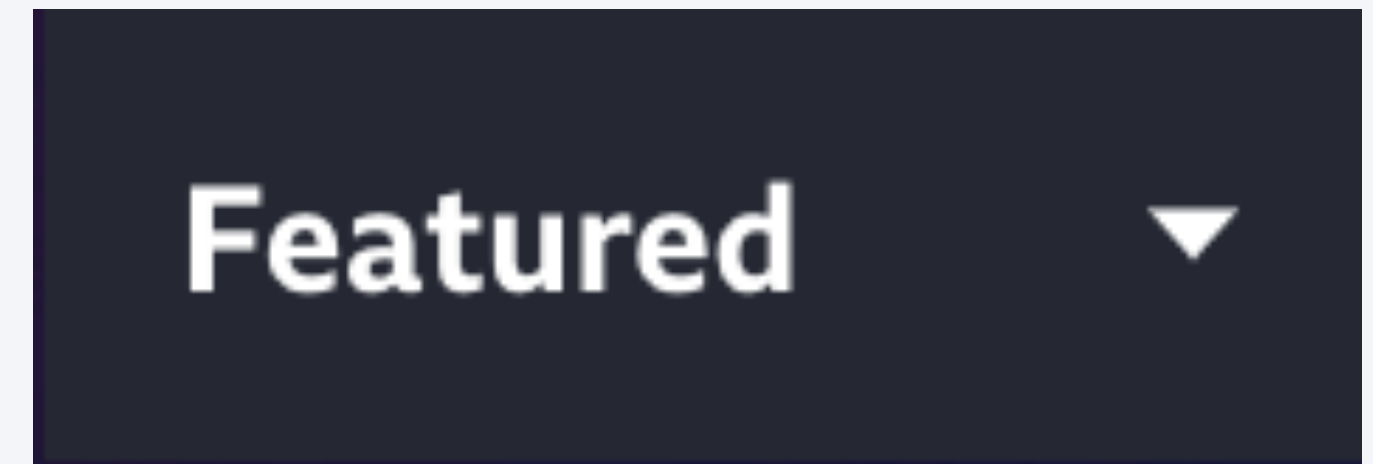


Responsive Image

- “alt” attribute on an `` tag is the alternative text for an image, shown when an image can't be loaded and read out by screen readers.
- No alt attribute on an image causes the image filename to normally be read out by a screen reader
- Empty alt text causes the image to normally be ignored by a screen reader as it's decorative
- Our component has an “alt” property which defaults to an empty string, rather than not present

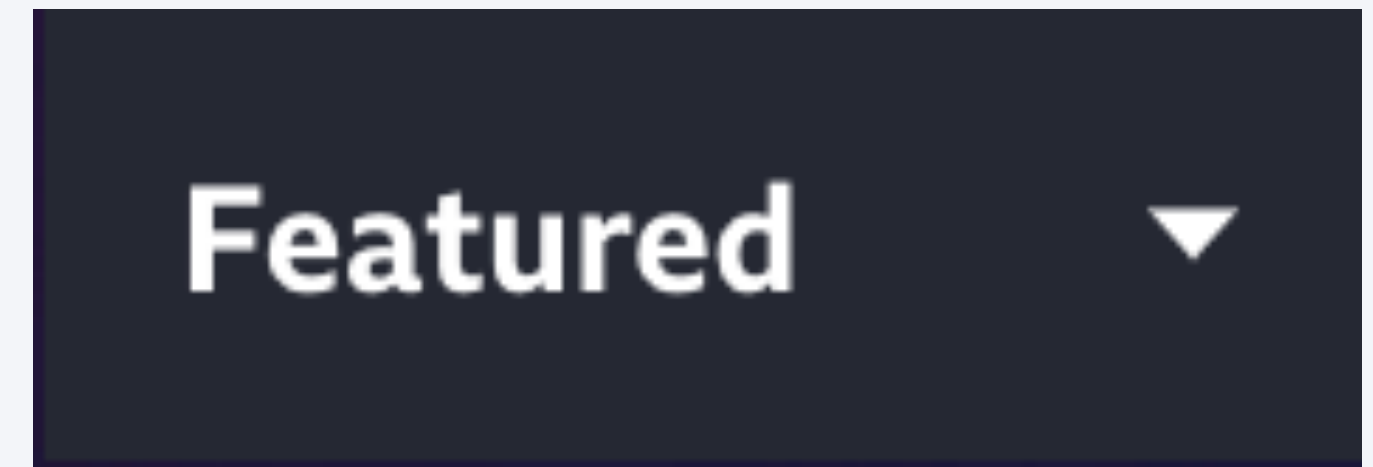
Dropdown

- Unlabelled dropdowns are difficult to understand without context
- “Featured, menu item main 1 item”



Dropdown


- We have a required “label” property which adds a `<label>` for the select
- “Featured, change sort, collapsed. Popup button main 1 item”





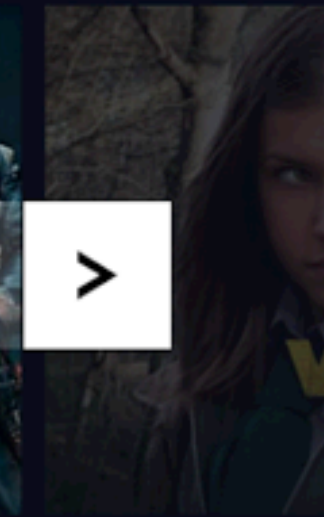


Section

- Is a `<section>` HTML element
- We have an “ariaLabel” property which adds an aria label
- This makes it a named landmark/region on the page for easier navigation
- This example would read out “region If you liked Doctor Who”

If you liked 'Doctor Who'



 <p>HIS DARK MATERIALS</p>	 <p>MERLIN</p>	 <p>ROBIN HOOD</p>	 <p>THE MUSKETEERS</p>	 <p>WOLF BLOOD</p>
Drama His Dark Materials: Series 1 One girl will change worlds	Drama Merlin: Series 1-5 Magic, adventure and the rise of King Arthur	Drama Robin Hood: Series 1-3 The legend, but not as you know it	Drama The Musketeers: Series 1-3 Swashbuckling action	CBBC Wolfblood: Series 1-3 From the start

Automated Testing

**“As designers, we
disable people when we
don’t get it right”**

Jamie Knight – Senior Accessibility Specialist, BBC

Automated Testing

With automated testing tools, we can more easily spot some of the times when we haven't got it right.

Google Lighthouse

Google Lighthouse

- Lighthouse is an automated tool for checking the quality of web pages
- Powered by aXe-core rules engine
- You can run it against any web page from Chrome Developer Tools
- It has audits for performance, accessibility, progressive web apps, and more.



What does Google Lighthouse test?

- Background and foreground colors have sufficient contrast ratio



- “Background and foreground colours do not have a sufficient contrast ratio. Low contrast text is difficult or impossible for many users to read”

What does Google Lighthouse test?

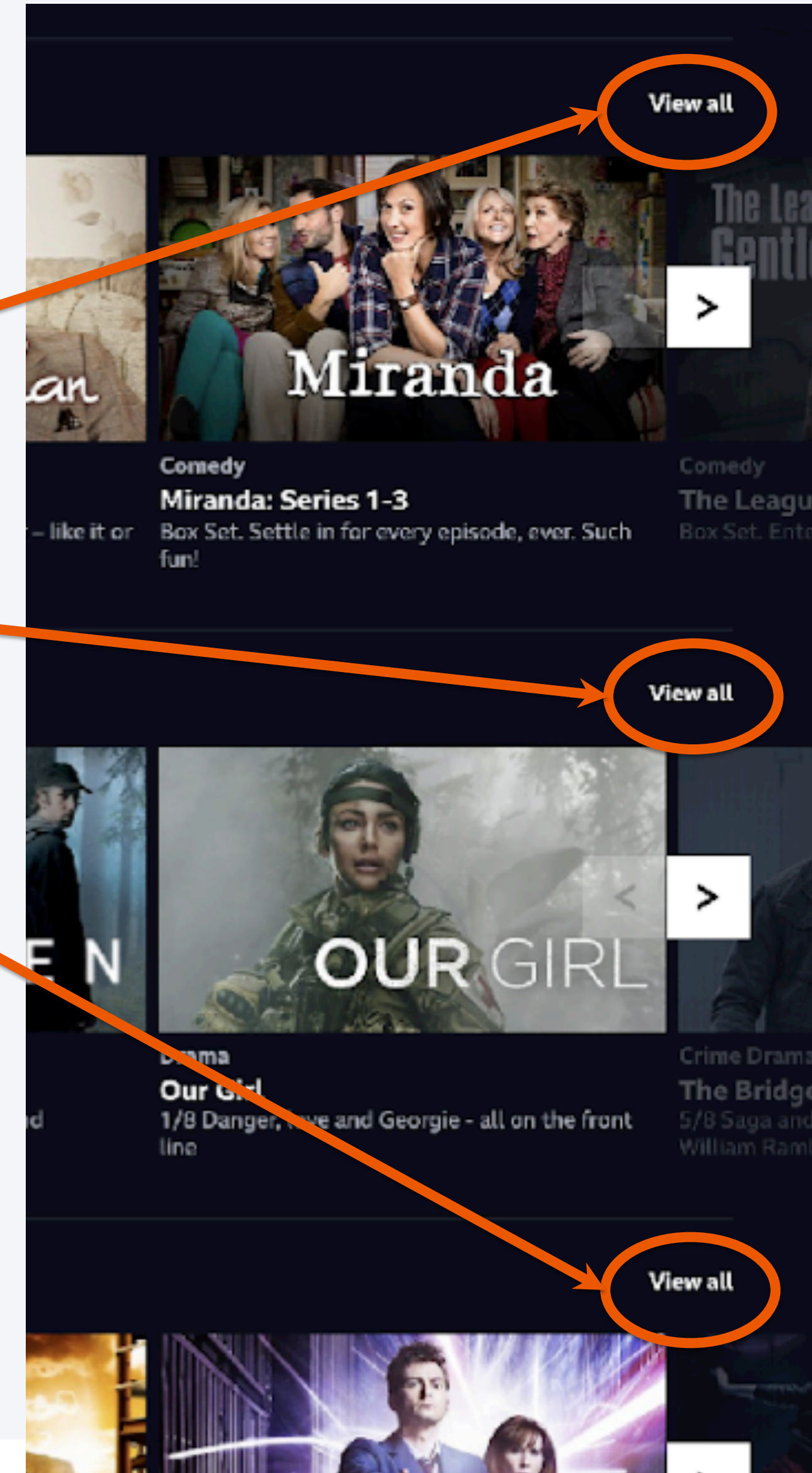
- Buttons have an accessible name



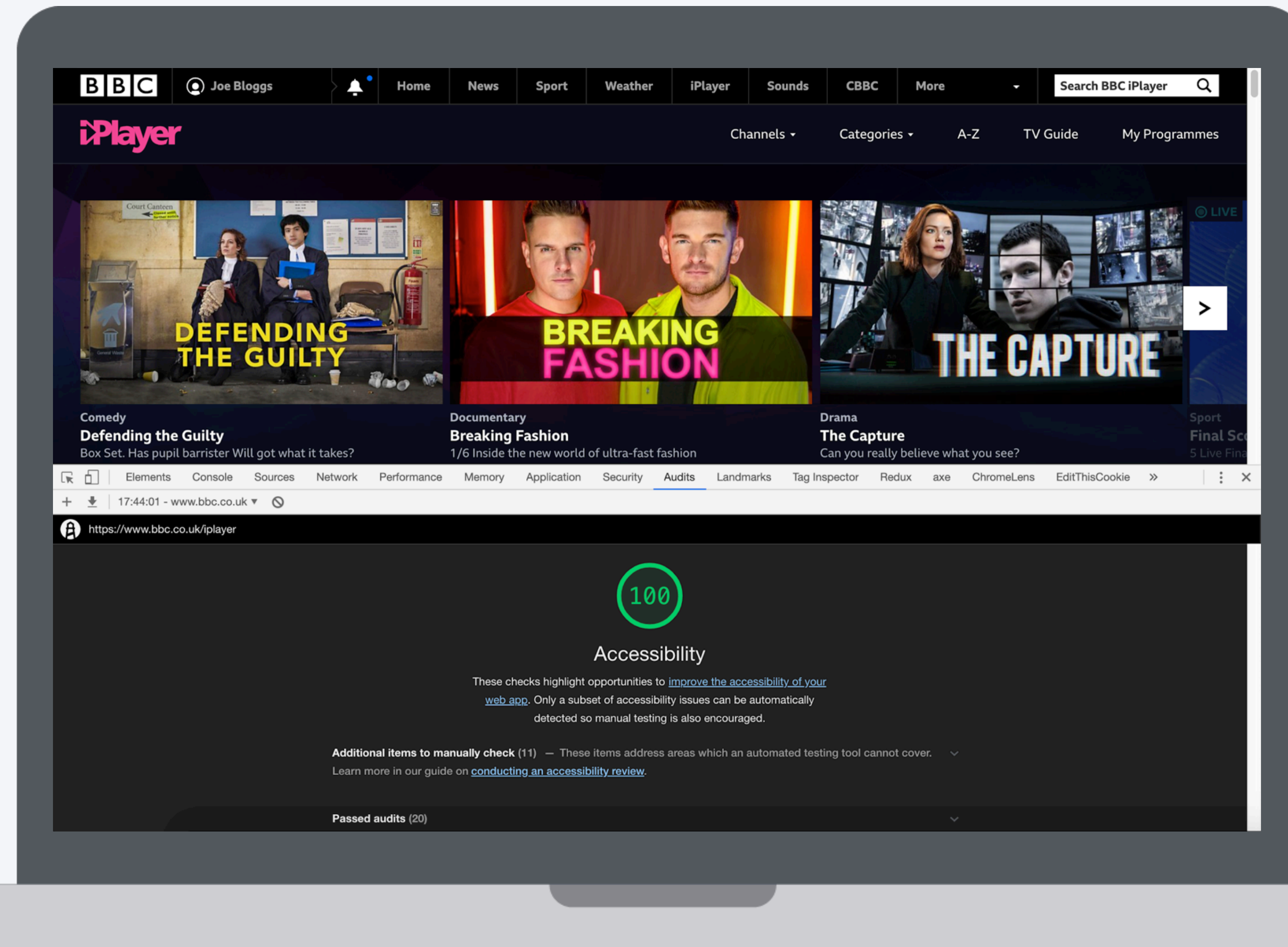
- “When a button doesn’t have an accessible name, screen readers announce it as ‘button’, making it unusable for users who rely on screen readers”

What does Google Lighthouse test?

- Links have a discernible name
- “Link text that is discernible, unique and focusable improves the navigation experience for screen reader users”



Running Google Lighthouse



bbc-a11y

What is bbc-a11y?

- BBC Accessibility Standards Checker
- An internally-developed tool
- Checks whether webpages meet the BBC's accessibility standards

The screenshot shows the GitHub repository page for 'bbc / bbc-a11y'. At the top, it displays the repository name and statistics: 14 watches, 22 stars, and 2 forks. Below this, there are tabs for 'Code', 'Issues (19)', 'Pull requests (5)', 'Projects (0)', and 'Insights'. The main content area features a header for 'BBC Accessibility Guidelines Checker' with a link to the website. Below the header, there are statistics: 427 commits, 7 branches, 31 releases, 10 contributors, and Apache-2.0 license. A navigation bar includes 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. The commit history table shows the following entries:

Commit	Message	Time
Latest commit b18bc8b	Tweak wording aimed at non-technical users	22 Mar
	Update ISSUE_TEMPLATE.md	7 months ago
	Upgrade electron to 1.7.12	4 months ago
	Allow arguments to docker image, fixes #243	2 months ago
	Add coverage table generator and update coverage.md	6 months ago
	Upgrade all dependencies	3 months ago
	scripts/generate-coverage	3 months ago
	Fix standard error	3 months ago
	scripts/generate-coverage	3 months ago
	Merge HTML tests into MAG guidelines structure	6 months ago
	Reorganise into shallow directory structure (#145)	a year ago
	Add mocha back to travis (#163)	a year ago
	v2.0.0 switches to the Mobile Accessibility Guidelines	6 months ago
	Use apache2 license and change author to BBC	7 months ago
	Tweak wording aimed at non-technical users	2 months ago

How to run `bbc-a11y`

Install npm (if you don't already have it)

1. Install the tool:

```
npm install bbc-a11y --global
```

2. Run the tool against a URL:

```
bbc-a11y http://www.bbc.co.uk
```

» bbc-ally <http://www.bbc.co.uk>

X <http://www.bbc.co.uk>

* Design: Content resizing: Text must be styled with units that are resizable in all browsers

- Text styled with px unit: //div[@id='bbcprivacy-prompt']/h2
- Text styled with px unit: //button[@id='bbcprivacy-continue-button']
- Text styled with px unit: //h1[@id='hp-bbc-homepage-content']

* Forms: Managing focus: Forms must have submit buttons

- Form has no submit button: //form[@id='orb-search-form']

⚠ Structure: Headings: Headings must be in ascending order

- First heading was not a main heading: //div[@id='bbcprivacy-prompt']/h2

* Text equivalents: Visual formatting: Use tables for data

- Table used for layout: //div[@id='orb-modules']/div[2]/section[13]/div[1]/div/div[1]

For details on how to fix these errors, please see the following pages:

- <http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile/design/content-resizing>
- <http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile/forms/managing-focus>
- <http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile/structure/headings>
- <http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile/text-equivalents/visual-formatting>

1 page checked, 5 errors found, 1 warning, 0 standards skipped

a11y-tests-web

What is a11y-tests-web?

- We've created a wrapper around `bbc-a11y` and Google Lighthouse.
- Allows you to run Google Lighthouse and/or `bbc-a11y` from one place

The screenshot shows the GitHub repository page for 'bbc / a11y-tests-web'. At the top, it displays the repository name, a navigation bar with 'Code', 'Issues 3', 'Pull requests 0', 'Insights', and 'Settings', and interaction buttons for 'Unwatch', 'Unstar 8', and 'Fork 1'. Below this, a description states 'Runs automated accessibility tests against configurable lists of webpages' with an 'Edit' button. A topic bar includes 'iplayer-web', 'accessibility', 'lighthouse', 'automated-testing', and 'bbc'. A summary bar shows '38 commits', '2 branches', '6 releases', and '4 contributors'. The main content area features a 'Branch: master' dropdown, a 'New pull request' button, and buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A commit history table follows, listing recent commits by user 'andysuk' with their descriptions and dates. The bottom section shows the 'README.md' file content, which includes the repository name 'a11y-tests-web', a 'build passing' status indicator, and a description: 'Uses `bbc-a11y` and Google Lighthouse to run a suite of automated tests to test accessibility across a set of webpages, defined in a config file.'

File	Description	Time
<code>.github</code>	Create issue_template.md	4 months ago
<code>config</code>	Remove page from iPlayer config that no longer exists	2 months ago
<code>lib</code>	Add and improve test coverage (#6)	4 months ago
<code>test</code>	Get iPlayer Web baseUrl from environment variable (#8)	2 months ago
<code>.gitignore</code>	Update .gitignore	4 months ago
<code>.travis.yml</code>	Create .travis.yml	4 months ago
<code>CONTRIBUTING.md</code>	Update CONTRIBUTING.md	4 months ago
<code>LICENSE</code>	Add LICENSE	4 months ago
<code>README.md</code>	Add libgconf-2-4 requirement to README (#7)	3 months ago
<code>buildBbcA11y.js</code>	Rename files and move logic around in preparation for adding tests	4 months ago
<code>cleanBbcA11y.js</code>	Rename files and move logic around in preparation for adding tests	4 months ago
<code>package-lock.json</code>	3.1.0	2 months ago
<code>package.json</code>	3.1.0	2 months ago
<code>runLighthouse.js</code>	Add lighthouse and change config directory structure (#2)	4 months ago

How to run a11y-tests-web

Install npm (if you don't already have it)

1. Install the package globally:

```
npm install @bbc/a11y-tests-web --global
```

1. Run the command anywhere from your command-line, e.g.

```
A11Y_CONFIG=iplayer-web/all a11y-tests-web  
lighthouse -m junit-headless
```

How to run a11y-tests-web

You could also make it part of your Node application's scripts.

1. Add the package to your application's dev dependencies:

```
npm install @bbc/a11y-tests-web --save-dev
```

1. Add a line to your application's scripts e.g.

```
"test:a11y": "A11Y_CONFIG=iplayer-web/app-playback-test a11y-tests-web bbc-a11y -m interactive"
```

**“Testing shows the presence,
not the absence of bugs”**

Edsger W. Dijkstra

Manual Testing

Manual testing with `bbc-a11y`

```
bbc-a11y http://www.bbc.co.uk --manual
```

Manual testing with bbc-a11y

The screenshot shows the BBC iPlayer website interface. At the top, there is a navigation bar with the BBC logo, a 'Sign in' button, and a search bar labeled 'Search BBC iPlayer'. Below this is a secondary navigation bar with the iPlayer logo and links for 'Channels', 'Categories', 'A-Z', 'TV Guide', and 'My Programmes'. The main content area features a grid of program thumbnails. The first row includes 'Rugby League: Challenge Cup' (Sport), 'All Girls Live at the Apollo' (Comedy), and 'Britain's Forgotten Men' (Documentary). The second row, under the heading 'Box Sets', includes 'The Split', 'We Can Be Heroes', 'Our Girl', and 'Fleabag'. A blue testing overlay is positioned at the bottom of the page, containing the question: 'Is information conveyed with colour also identifiable from context or markup?'. Below the question is a text input field labeled 'Comments' and two buttons labeled 'Yes' and 'No'.

BBC a11y

BBC Sign in

News Sport Weather iPlayer TV Radio More

Search BBC iPlayer

iPlayer Channels Categories A-Z TV Guide My Programmes

LIVE

Sport **Rugby League: Challenge Cup**
2018: Quarter Final: St Helens...

Comedy **All Girls Live at the Apollo**
1/2 Top female comedy talent

Documentary **Britain's Forgotten Men**
Box Set. Ignored? Abandoned? Meet the white working class

Drama **A Very**
1/3 Power story

Box Sets

THE SPLIT

WE CAN BE HEROES

OUR GIRL

FLEABAG

Is information conveyed with colour also identifiable from context or markup?

Comments



Yes No

Manual testing with bbc-a11y

Asks questions including:

- Is information conveyed with colour also identifiable from context or markup?

#	Country				
1	 NOR	14	14	11	39
2	 GER	14	10	7	31
3	 CAN	11	8	10	29
<hr/>					
19	 GB	1	0	4	5

#	Country				
1	 NOR	14	14	11	39
2	 GER	14	10	7	31
3	 CAN	11	8	10	29
<hr/>					
19	 GB	1	0	4	5

Manual testing with bbc-a11y

Asks questions including:

- Is information conveyed with colour also identifiable from context or markup?
- Do all colour combinations pass a colour contrast check?
- Do all text links have a mouse over state change?
- Do all focusable elements have a clearly identifiable visual style when they have focus?
- Does the page function without JavaScript or CSS?
- Is the content visible and usable with the page zoomed to 200% of normal?
- Is the content visible and usable with the text resized to 200% of normal?

Screen Reader Testing

Every new feature/page is tested by our QA team with

- NVDA on Windows
- VoiceOver on Mac

Team Culture

Knowledge Sharing

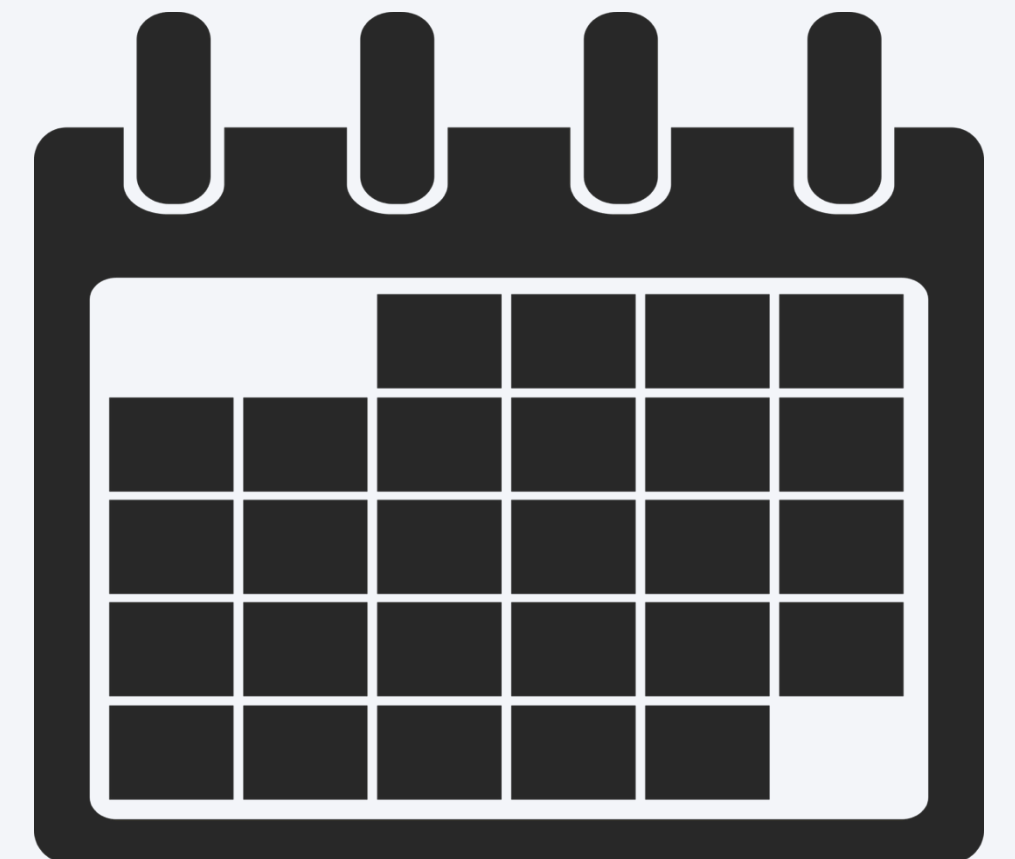
- We run cross-discipline accessibility training sessions e.g.
 - Introduction to accessibility
 - screen reader training
 - accessibility basics for web developers
 - deaf awareness



Knowledge Sharing

iPlayer & Sounds accessibility guild have fortnightly catch-ups to:

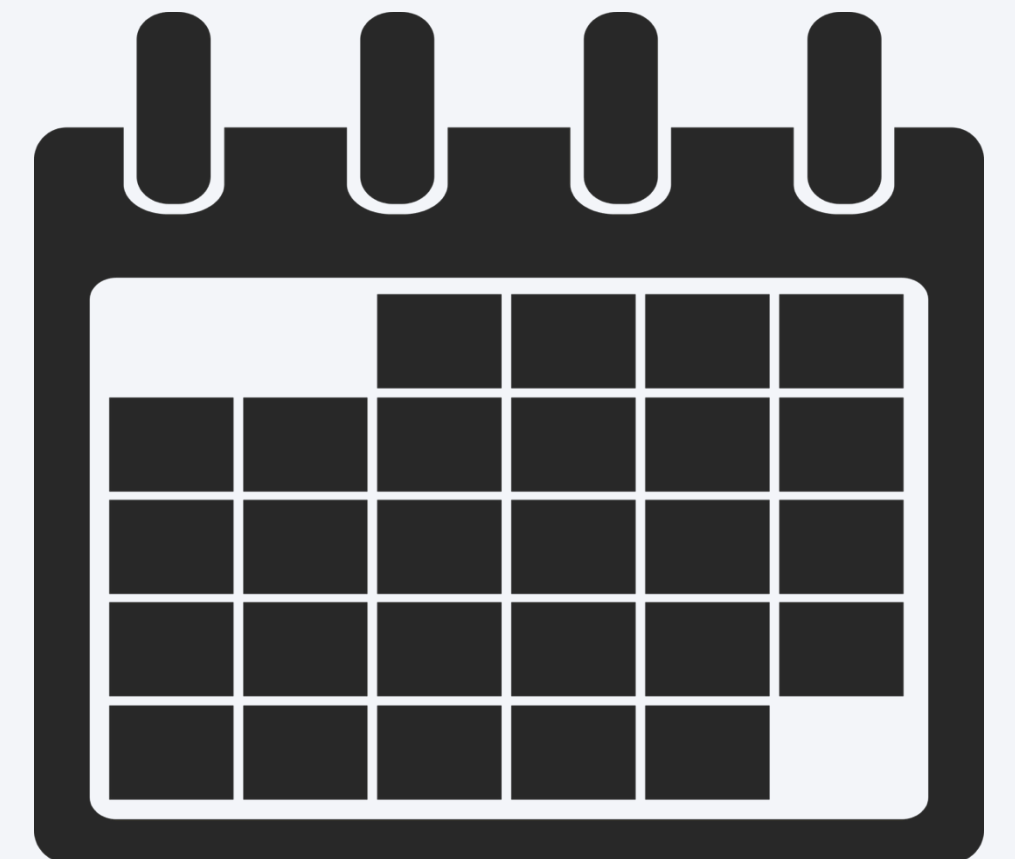
- Discuss and fix any on-going issues
- Discuss and implement ideas for accessibility improvements
- Share new accessibility related knowledge



Knowledge Sharing

We also have cross-BBC accessibility champion meet-ups which include:

- Talks from external speakers
- Lightning Training Sessions



**“The power of the Web is in
its universality.
Access by everyone
regardless of disability is an
essential aspect”**

Tim Berners-Lee - Inventor of the World Wide Web

3 Key Things

1. Building accessibility in from the start
2. Testing
3. Team Culture

Thanks!



Andy Smith
@AndyMSUK



Emily Atkins
@Em_Atkins

BBC-A11Y: github.com/bbc/bbc-a11y

A11Y Tests Web: github.com/bbc/a11y-tests-web

Home Office Posters: github.com/UKHomeOffice/posters

BBC Accessibility Guidelines: bbc.co.uk/accessibility/forproducts/guides/mobile/