MRS Awards 2022 Finalist Business-to-Business Research

MRS Awards 2020 Winner Innovation of the Year

MRS Awards 2019 Winner Global Agency of the Year

MRS Inclusion Pledge

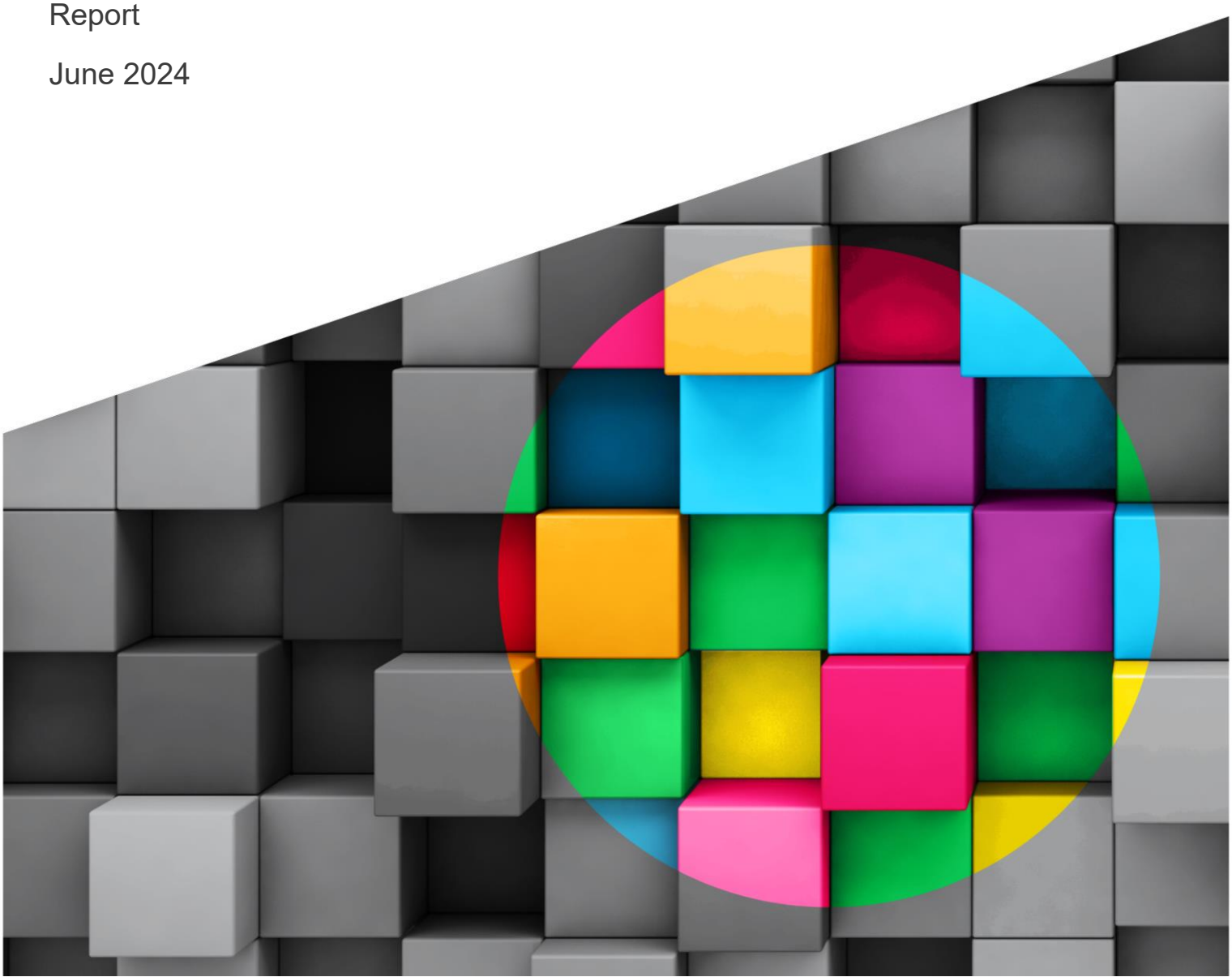
STRAT 7

JGSW

Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines

Report

June 2024



Contents

1. Executive summary

1.1. Background

On 10 June 2022, the CMA published the final report of its market study into 'mobile ecosystems' including mobile phone operating systems, app stores and web browsers[1].

Following a period of consultation, the CMA launched a market investigation on Tuesday 22 November 2022 into the supply of mobile browsers and mobile browser engines, and the distribution of cloud gaming services through app stores on mobile devices.

This qualitative research has been commissioned by the CMA as part of its market investigation. The research has the aim of seeking to understand the experiences of a wide range of web developers, including those who might not ordinarily engage with the CMA.

Browsers are used to access and navigate the internet and are a key gateway between users and businesses with an online presence. In this sense, they are, together with native apps[2], the main way in which consumers engage with digital content on mobile devices. The two most used browsers are Apple's Safari and Google's Chrome, which have a combined share of supply of around 90% in the UK as of 2023.[3]

Browsers run on browser engines. Browser engines transform web page source code into web pages (or web apps) that users can see and engage with, and are fundamental to the speed and capability of browsers. As of March 2024, at least 97% of browsing on Android devices in the UK was powered by Google's Blink browser engine, while 100% of browsing on iOS devices in the UK was powered by Apple's Webkit browser engine.[4] Browsers also enable people to access 'web apps', which are websites with the potential to be more like native apps without being tailored to each operating system. Native apps can also display web content using an In-App browser (IAB) based on functionality enabled by the operating system (OS).

Both Google and Apple supply the two browser engines on which most mobile browsers are powered on their mobile ecosystems – the exception being Firefox which uses its Gecko engine on Android devices.

1.2. Research approach

The target audience for this research was web developers who develop for the UK market, with a minimum of two years' experience working in web development for mobile, specifically developing websites and web apps with a primary focus on optimising those for mobile devices. Within this, the

[1] See CMA mobile ecosystems case page for all case documentation including the full final report and a shorter summary of the final report: Mobile ecosystems market study - GOV.UK (www.gov.uk)

[2] Most mobile apps are 'native' i.e. written to run on a specific operating system. Some come pre-installed on mobiles, but the majority are downloaded through app stores.

[3] https://www.statista.com/statistics/489413/uk-market-share-held-by-mobile-online-browsers-by-month/

[4] Source: Cloudflare Radar

aim was to speak to a mix of front-end[5] and full-stack developers[6] , working for a range of different clients, and in a mix of types of roles, including those working freelance, in agencies or in-house.

The target number of interviews for this study was 60 web developers. Fieldwork ran from 13th February 2023 – 31st March 2023. Participants were free found, recruited primarily from LinkedIn.

In March 2023, the market investigation was suspended pursuant to an order from the Competition Appeal Tribunal (CAT). This ruling meant that the research project was halted. At this point, 47 out of 60 interviews had been completed and some initial analysis had begun.

On 24 January 2024, the market investigation recommenced following a lifting of the original CAT order. The CMA requested that the research re-commence. Follow-up interviews took place from 4th March 2024 – 25th March 2024 and involved 13 of the individuals originally interviewed in 2023 to check whether anything had changed in their behaviours or views since their original interview.

Overall, the re-convened interviews confirmed that the data collected in 2023 was still valid, and remained relevant to inform the CMA's research objectives. The only key difference noted was that AI was increasingly influencing many participants' choices and behaviours.

Qualitative research focuses on understanding underlying issues, experiences, and motivations rather than providing percentages or numbers. The findings are indicative of the market rather than representative. They are based on participants' recall, understanding, and interpretation.

1.3. Summary of findings: Web developer attitudes and behaviours

As context, it is helpful to understand the key attitudes and behaviours of the web developers interviewed.

Personas

Web developers interviewed in the research tended to fall into two subgroups or 'personas' that can be described as: 'Innovators', who were more outward looking and knowledgeable, and made up around a third of the participants in this research, and 'Nine-to-fivers' who were more focused on simply getting their job done. These proportions might not be representative of the proportions of these groups in the wider web developer population.

Development process

Web developers described the process of building web apps and websites for mobile as building once then checking for compatibility across browsers, operating systems and devices. Ongoing web app and website maintenance also formed a significant part of their work.

[5] Front-end developers focus on the functionality of a site including the user interface (as opposed to back-end developers who focus on the infrastructure of a site including web hosting and security).

[6] Full-stack developers work on both front-end and back-end.

1.4. Summary of findings: Industry trends and challenges

Web developers who participated in the research were asked about the underlying industry trends and challenges that impact on their work.

Trends

A key trend that web developers mentioned was the increased ease of web app and website development for mobile devices over the last five years. Developers mentioned a range of factors driving this:

- Use of frameworks[7] and libraries[8], such as React, Angular, Node, Vue, Bootstrap, WordPress and Solid, where re-use of modules of code can reduce manual changes required to build and ensure compatibility across browsers and devices.
- Increasing use and capabilities of web apps making it easier and faster to deploy and manage apps compared to native apps, as you build once (rather than for each specific operating system), and also can make instant updates as needed.
- Improving functionality and standardisation of browsers, meaning fewer compatibility issues arise.
- AI tools bringing speed to some tasks such as writing, editing, simplifying, and annotating code.
- Many communities and resources for advice and troubleshooting e.g. Stack Overflow and code repositories like GitHub.

Challenges

Though developing web apps and websites for mobile devices is getting easier, there are still some common day-to-day technical challenges. Key challenges stemmed from the existence of multiple mobile screen sizes, multiple versions[9] and updates to ecosystems and tools, and the need to maintain privacy and security – ensuring browser compatibility is a part of these challenges but was not a top-of-mind concern.

[7] A web development framework is a software framework that is designed to support the development of web applications. Web frameworks provide a standard way to build and deploy web applications. Many web frameworks provide libraries, templating frameworks, and they often enable code reuse. They provide a foundation to build upon, offering a generic functionality that can be selectively modified by additional code written by the web developer.

[8] Libraries are collections of pre-written code that developers can incorporate into their projects. Libraries enable code reuse in a modular fashion.

[9] Multiple versions (and updates) referred to by participants included new product releases, updates to fix security issues or bugs, or, in a few cases, switches to more up-to-date types of services, such as a different kind of framework with better functionality. This led to the need to develop for multiple versions of operating systems, browsers, search engines, SEO (search engine optimisation) policies, tools and frameworks, APIs, plug-ins, libraries and so on.

1.5. Summary of findings: Browsers and browser engines

Browser perceptions and choice

Choice of the main browser to prioritise and optimise for first was driven by a range of factors, and the features of the browser were not the only or main influence.

For most participants[10] the main browser they optimised for was Chrome driven by its share of web traffic, familiarity, ease of use and the quality of the developer tools. A few favoured Safari, as they saw it as better for developing on Apple hardware that they used, and a few others (who tended to be 'Innovators') preferred use of Firefox and Brave as they favoured these for privacy and security as both an end-user and developer.

Browser engine perceptions

Detailed points of view about browser engines were not widespread. Participants, especially 'Nine-to-fivers', were more likely to refer in general terms to browsers, rather than mentioning specifics about how browsers differed depending on their browser engine.

Browser issues and restrictions

The number of browser-related issues encountered during development of web apps and websites for mobile devices was seen to be reducing. Participants tended to describe browsers and browser engines as having differences, or benefits and drawbacks, rather than reporting significant issues or restrictions.

More mentioned differences when developing for Safari than for Chrome or other browsers. A few specified that it was not possible to install extensions on WebKit-based browsers but this was feasible on some Chromium-based browsers; a few others mentioned that WebKit-based browsers required different formats e.g. for video and that Apple was generally slower than Google to introduce new features to browsers.

1.6. Summary of findings: Compatibility with different browsers

Compatibility process

Web developers tended to test the compatibility of their web apps and websites for mobile devices against browsers with the biggest market share, namely Chrome, Safari, sometimes Firefox, Brave or Edge, but rarely Internet Explorer now as its use has significantly declined and it is also considered difficult to develop for.

The level of effort put into ensuring compatibility was determined by the client brief and resource available. Testing compatibility commonly involved using simulators such as BrowserStack. For some projects, some stated that they only undertook basic manual checks on available phones. A few mentioned more involved procedures such as QA (quality assurance) departments undertaking tests or automated testing across multiple physical devices by specialist third parties.

[10] The term participants refers to the web developers who participated in the research interviews.

Resource to ensure compatibility

Most web developers interviewed felt ensuring compatibility across browsers was a small part of their work. Most estimated that it typically took 5-10% of their time, however some estimated the time taken was outside this range, with a few saying it took very little or even a negligible amount, and a few others that it took 20-25% of their time. Most participants reported use of frameworks and libraries when developing web apps and websites for mobile devices, re-using modules of existing code rather than 'building from scratch', which has helped to save time on checking compatibility across browsers and devices. Increasing standardisation across browsers was also leading to less work to ensure compatibility, as fewer issues were arising.

Compatibility challenges

Compatibility issues mentioned by participants tended to relate to minor adjustments that were needed to ensure a web app or website worked well across browsers, rather than major concerns or limitations on what was developed.

These issues related to 'differences' between the browsers, or on few occasions 'bugs' or more complex functionality workarounds that were needed. No single issue was mentioned by more than a few people. Innovators sometimes cited several issues, whereas a few Nine-to-fivers mentioned none.

Examples of compatibility issues given related to minor adjustments to layout, ensuring more visual, complex features work on different devices e.g. animations, videos, streaming, sliders, and ensuring access to different mobile phone features e.g. camera access.

1.7. Summary of findings: iOS and Android ecosystems

Overview

When participants thought about 'ecosystems' they would often include many elements in their assessment, including the main browser and browser engine used on that system, the operating system, devices, the relevant app store, and general views of the company (or companies). So participants' comments about the iOS and Android ecosystems replicate some of the comments that they made about browsers and browser engines, for example, iOS being seen as slower than Android to allow new functionality and features, as well as Safari being seen as slower than Chrome.

Overall, most participants did not see any of the negatives or weaknesses in either ecosystem as barriers that would stop them developing for either of the ecosystems.

Many of the strengths and weaknesses perceived in Android or iOS ecosystems seemed to be a reflection of that ecosystem's approach to how it managed its services, i.e. Android was more open compared to iOS, which was seen as more closed or a 'walled garden'. Each of these approaches was seen to have both weaknesses and strengths that reflected this overall 'philosophy', and some participants would develop first, or prefer to develop, for the ecosystem that best matched their personal preferences for such systems.

The iOS ecosystem

The iOS ecosystem was seen positively in regard to security, privacy and stability. Some also saw it as the best end-user experience and a minority saw it as the best developer experience.

Conversely it was also seen as slower (to improve or add more functionality) and more restrictive than Android, with a minority feeling that iOS policies restricted what they (as developers) could do within

the ecosystem – meaning they had less ability to innovate and/or it took more time to develop in (as they had to spend time on workarounds to overcome restrictions).

The Android ecosystem

A greater number of participants stated they were more familiar with the Android ecosystem than iOS. Familiarity was seen as a major benefit of using Android by users.

Frequent updates and improvements, and a strong set of developer tools (which in turn lead to greater flexibility) were seen as benefits by many participants. The Android ecosystem was overall seen as easier to use than iOS. It was also seen by most as being very widely used by both developers and clients.

However, limitations were perceived by a minority in relation to security and privacy. For a few, device compatibility was also an issue, given the wide range of Android devices on the market, particularly in regard to screen sizes (although it was also mentioned that the impact of this issue had reduced over time because of the use of developer tools and frameworks).

1.8. Summary of findings: Future outlook – AI and other changes

Future trends

AI, as noted above, was seen as the main trend that would impact significantly on web development in the future. ChatGPT was being used by a few web developers in 2023 before the project was halted, and AI tools in general had become more widely used among those who took part in follow-up interviews in 2024. Most participants saw AI as already making their 'day jobs' easier and/or having the potential to do so in the future, including helping to ensure the compatibility of their websites and web apps with different browsers. A minority felt that it may impact negatively on jobs (because fewer developers would be needed) and the industry in general (because clients would have access to better self-service web packages).

Other future trends identified were continuations of existing trends (such as the continued adoption of new tools, libraries and so on, or the continued move towards mobile first). A small number of participants specifically felt that ensuring compatibility between browsers would continue to get easier, driven by the ongoing release of the new tools and AI.

Competition and choice in the future

Towards the end of the interview, participants were directly asked about whether more choice around browsers and browser engines would be beneficial or disadvantageous for them as web developers.

Overall, the majority of participants (both Nine-to-fivers and Innovators) did not want more choice of browser or browser engines in future as they felt this would cause more work for them. They seemed broadly happy with the status quo as it stood at the time of their interview. A small minority actively wanted more choice across browsers and browser engines as they perceived this may bring more flexibility, competition, and innovation.

2. Background, objectives and approach

2.1. Background to the research

On 10 June 2022, the CMA published the final report of its market study into 'mobile ecosystems' including mobile phone operating systems, app stores and web browsers[11].

Following a period of consultation, the CMA launched a market investigation on Tuesday 22 November 2022 into the supply of mobile browsers and mobile browser engines, and the distribution of cloud gaming services through app stores on mobile devices.

This qualitative research has been commissioned by the CMA as part of its market investigation. The research has the aim of seeking to understand the experiences of a wide range of web developers, including those who might not ordinarily engage with the CMA.

The browser and browser engine market

Browsers are used to access and navigate the internet and are a key gateway between users and businesses with an online presence. In this sense, they are, together with native apps[12], the main way in which consumers engage with digital content on mobile devices. The two most used browsers are Apple's Safari and Google's Chrome, which have a combined share of supply of around 90% in the UK as of 2023.[13]

Browsers run on browser engines. Browser engines transform web page source code into web pages (or web apps) that users can see and engage with, and are fundamental to the speed and capability of browsers. As of March 2024, at least 97% of browsing on Android devices in the UK was powered by Google's Blink browser engine, while 100% of browsing on iOS devices in the UK was powered by Apple's Webkit browser engine.[14] Browsers also enable people to access 'web apps', which are websites with the potential to be more like native apps without being tailored to each operating system. Native apps can also display web content using an In-App browser (IAB) based on functionality enabled by the operating system (OS).

Both Google and Apple supply the two browser engines on which most mobile browsers are powered on their mobile ecosystems – the exception being Firefox which uses its Gecko engine on Android devices.

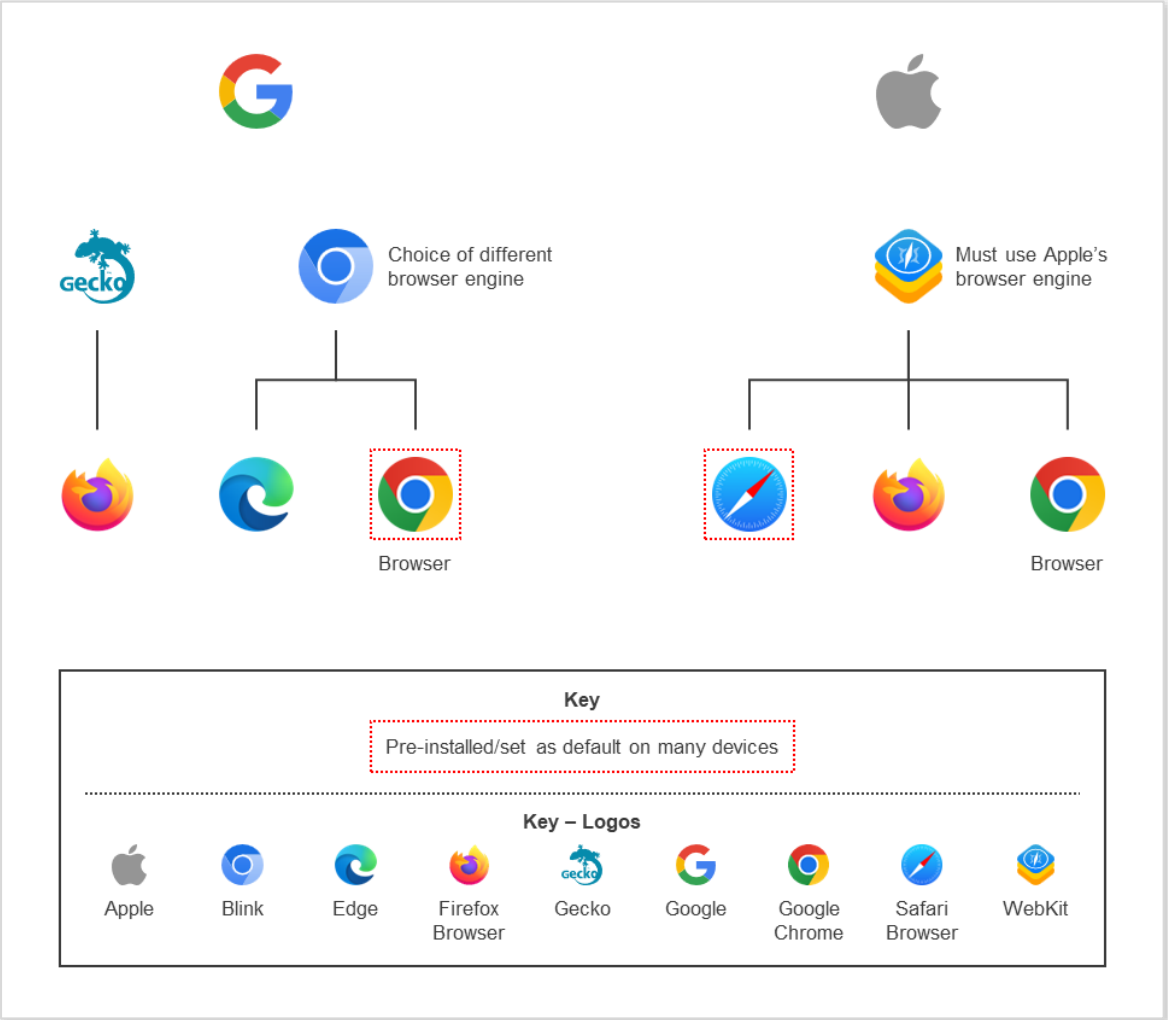
[11] See CMA mobile ecosystems case page for all case documentation including the full final report and a shorter summary of the final report: Mobile ecosystems market study - GOV.UK (www.gov.uk)

[12] Most mobile apps are 'native' i.e. written to run on a specific operating system. Some come pre-installed on mobiles, but the majority are downloaded through app stores.

[13] https://www.statista.com/statistics/489413/uk-market-share-held-by-mobile-online-browsers-by-month/

[14] Source: Cloudflare Radar

Figure: Browser engine choice within each ecosystem



Key terminology

Mobile device**s:** smartphones and tablets.

Mobile browsers: applications which enable users of mobile devices to access the world wide web. Browsers are a crucial gateway for people to access the web and a key gateway between users and businesses. The two most used browsers are Apple's Safari and Google's Chrome.

In-app browsers: the function by which users access web pages in native apps which are not a dedicated browser.

Browser engines: browsers run on browser engines which are fundamental to the speed and capability of browsers. Mobile browser engines are the underlying technology which browser applications on mobile devices use to transform web page source code into content with which users can engage with.

Native apps: are pieces of computer software developed for use on a particular mobile OS or device. They interact directly with the operating system to offer additional features and functionality.

Web apps: Apps accessible via web browsers like a regular webpage which do not need separate versions to be coded for separate operating systems. Web apps have more functions compared to traditional webpages, including opportunities for interactions and they can partially operate offline. Some web apps (like progressive web apps) are designed in a way that makes them as close as possible to native apps in terms of look and feel and user experience.

Web compatibility: is the browser's ability to properly access and display the content on a particular web page.

Framework: is a software framework that is designed to support the development of web applications. Web frameworks provide a standard way to build and deploy web applications. Many web frameworks provide libraries, templating frameworks, and they often enable code reuse. They provide a foundation to build upon, offering a generic functionality that can be selectively modified by additional code written by the web developer.

Libraries: are collections of pre-written code that developers can incorporate into their projects. Libraries enable code reuse in a modular fashion.

2.2. Research objectives

The central research objective was to further develop the CMA's understanding of the experiences of web developers in relation to mobile browsers and mobile browser engines. More specifically, the research needed to explore:

- The experiences of a range of web developers in developing websites and web apps optimised for mobile;
- Web developers' experiences specific to iOS and also Android;
- What web developers saw as the key parameters that distinguish browsers and browser engines;
- The extent to which web developers valued specific aspects of browsers (such as following standards, or including new features);
- Whether web developers have faced any barriers with browsers and browser engines when developing websites and web apps;
  - If so, what barriers have they faced and what has been the impact of these?
- Whether and to what extent Apple and Google policies limited web developers' offering in the websites and web apps they develop in terms of capability/performance/differentiating characteristics; and
- The extent to which web developers' approach to developing websites and web apps was impacted by the presence of the restriction on third-party browser engines on iOS.

2.3. Research target audience

The target audience for this research was web developers who develop for the UK market.

The aim was to ensure the following were covered in the interview sample:

- Minimum of two years' experience working in web development for mobile, specifically developing websites and web apps with a primary focus on optimising for mobile devices.
- Web developers who would not ordinarily engage in government consultations to ensure a wide range of developers beyond those highly engaged in this area.
- Mix of types of developer:
  - Front-end developers who focus on the functionality of the site itself including the user interface.
  - Full-stack developers who work on both front-end and back-end.
  - (Back-end developers who only focus on the infrastructure of a site including web hosting and security were not included).
- Mix of employment situations:
  - In-house experience – from large businesses e.g. banking, publishing and broadcasting and tech businesses that have in-house web development services.
  - Agency and self-employed experience – as web development is a fragmented sector with no major players.
- Mix of preferences for Apple ecosystem, Google ecosystem or no preference.
- Developers working on projects for a range of clients by sector and size.

  Key client sectors:

  - Advertising/digital marketing/market services agencies.
  - Social media.
  - Public relations and communications businesses.
  - Internet publishing and broadcasting.
  - E-commerce and online auctions.
  - Gaming for mobile.

  Range of client size:

  - Micro (1-9 staff), small (10-49), medium (50-249) and large businesses (250+).

2.4. Research method

An overview of the research process is set out below:

1. Set up	2. Recruitment	3. Fieldwork	4. Analysis
Immersion and set up	Screening and recruiting	Main fieldwork	Interim analysis
Sample design	Scheduling and pre-task	Follow-up fieldwork	Final Reporting

1. Set up

Immersion and set up: The Jigsaw project team of five interview moderators received a detailed briefing from the CMA on the sector and background to the research. This enabled Jigsaw to design the research to take account of the complexities, ensure we would speak to a good cross-section of people indicative of web developers of websites and web apps for mobile, and have an open and unbiased conversation about their views and experiences.

Sample design: The CMA wanted to speak to a wide range of web developers, and specifically those who had not engaged with the CMA's market investigation through any other means nor the earlier market study. It was agreed with the CMA that respondents would be free found by Jigsaw's specialist recruitment partner Roots Research Limited. A target of 60 online interviews with developers was set with the CMA, with minimum quotas on different developer roles, business types and development preferences to ensure a good mix of participants.

2. Recruitment

Screening and recruiting: A screener was developed and agreed with the CMA to be used by Roots Research to identify the correct individuals for interview. Participants were recruited through open sources, primarily LinkedIn. A small number of interviewees (15) were also recruited from the recruiter's existing database of 450,000 individuals from around the UK who had already expressed interest in taking part in research studies. *A copy of the recruitment screener is available in Appendix section 4.1.*

Scheduling and pre-interview task: Following recruitment, online interview appointments were scheduled, and participants were sent a light-touch pre-task before their interview. The pre-task was not compulsory but designed as a prompt to encourage participants to start thinking about the interview topic more consciously. The pre-task encouraged respondents to reflect on what browser and browser engine features, functionalities or APIs help or hinder them when developing for mobile devices. *A copy of the pre-task is available in Appendix section 4.2.*

3. Fieldwork

Main fieldwork: A discussion guide was agreed with the CMA. Fieldwork ran from 13th February 2023 – 31st March 2023. The CMA listened in to 5 initial interviews, as an informal pilot, leading to some small refinements to the discussion guide. In March 2023, the market investigation was suspended pursuant to an order from the Competition Appeal Tribunal (CAT). This ruling meant that the research project was halted. At this point, 47 out of 60 interviews had been completed and some initial analysis had begun.

Follow-up fieldwork: On 24 January 2024 the market investigation recommenced following a lifting of the original CAT order. The CMA requested that the research re-commence. The CMA decided that the priority was to recontact some previous participants rather than reaching out to new developers. For the main fieldwork, the original goal was to speak to more than one developer in some larger businesses, to get both strategic and tactical viewpoints, however, when the project was halted only one set of 'paired' interviews in the same business had been undertaken. The CMA felt that most value would be gained by following up on whether anything had changed among some of those already interviewed rather than seeking new or new paired interviews given the wide range of types of developer spoken to and level of detail already gathered before the market investigation was halted.

The discussion guide for the follow-up interviews was agreed with the CMA and focussed on the same topics as the main stage, but with probes on what, if anything, had changed in the past year. In particular, it added extra questions on the use of AI, as AI tool ChatGPT was emerging at the end of the main fieldwork in 2023 as an influence on developer behaviours. Roots research recontacted all previous participants, to check that they still worked as developers for web apps and websites for mobile devices and to ask them to take part in a follow-up interview. Follow-up interviews took place from 4th March 2024 – 25th March 2024 and involved 13 of the individuals originally interviewed in 2023. *Copies of the discussion guides from the main and follow-up fieldwork are available in Appendix sections 4.3 and 4.4.*

Overall, the re-convened interviews confirmed that the data collected in 2023 was still valid, and remained relevant to inform the CMA's research objectives. The key difference noted, as early comments from participants in 2023 had begun to suggest, was that AI was increasingly influencing many participants' choices and behaviours.

4. Analysis

Interim analysis: The interviews were audio or video recorded with permission from the participants and, along with other evidence such as notes, were uploaded and stored securely by Jigsaw for analysis. For analysis, Jigsaw set up a Qualitative Analysis Frame in Excel with a row for each participant and a column for each key theme that the individual moderators filled in upon completion of their interviews. This meant the evidence around each topic was collated systematically, was grounded in the views of participants, and no voice was lost. Analysis was carried out collectively with input from all moderators involved in interviewing. Jigsaw shared interim headline findings with the CMA on 23rd March 2023 before the project was paused.

Final reporting: Once all fieldwork was complete, Jigsaw presented overall findings based on the main and follow-up interviews to the CMA in a PowerPoint debrief presentation. This report follows on from the debrief with further iteration and detail based on ongoing analysis of the data.

2.5. Sample structure

The goal was to include a wide range of participants indicative of those who develop website and web apps for mobile devices. A good spread of interviews was achieved, for the main stage and follow-up interviews.

Quotas

As mentioned in the previous section, 47 interviews out of a target 60 had been achieved when the project was paused in March 2023. After this, an additional 13 follow-up interviews were carried out when the project was restarted in January 2024 to sense check if findings were still valid from the initial interviews and whether views and behaviours had changed in the year that had past, making a total of 60 interviews.

To ensure no key groups were missed, minimum quotas were set for developer role, type of employment and development preferences. These quotas had been easily met when the main project was paused. This meant that the profile of participants fell out naturally during recruitment suggesting it is relatively indicative of the market.

	Target	MAIN STAGE No. of interviews: Achieved before project paused (Feb-Mar 2023)	FOLLOW UP STAGE No. follow up interviews: Achieved after project restarted (Mar 2024)
Front-end	Minimum 15	16	5
Full-stack	Minimum 25	31	8
In-house developer	Minimum 5	27	5
Web agency	Minimum 5	7	2
Freelance	No quota	13	6
Prefer to work with Apple ecosystem (i.e. iOS operating system and Safari browser)	Minimum 5	9	3
Prefer to work with Google ecosystem (i.e. Android and Chrome)	Minimum 5	11	1
No preference	No quota	27	9
Total interviews	**60**	**47**	**13**

Soft quotas

As well as having minimum quotas on some key profile measures, a range of other profile information was gathered and monitored to ensure a good spread of participants.

Seniority	No. interviews
Junior – hands-on, but do not supervisor/manage others	24
Mid – hands-on and supervise/manage others	19
Senior – lead and manage a team of developers	4
Total	**47**

Years of experience developing web apps/websites for mobile	No. interviews
2-5 years	22
6-10 years	18
More than 10 years	7
Total	**47**

Main focus in developing for mobile in last 2 years	No. interviews
Mainly websites	16
Mainly web apps	14
Both	17
Total	**47**

Size of business develop mobile web apps/websites for (multiple answers)	No. mentions
Large (250+ employees)	28
Medium (50-249 employees)	32
Small (10-49 employees)	31
Micro (1-9 employees)	19

Client sectors develop mobile web apps/websites for (multiple answers)	No.mentions
Marketing/communications/advertising/PR agencies	20
Social media	10
Internet publishing and broadcasting	11
Ecommerce and Online auctions	24
Gaming for mobile	2
Other	14

2.6. Methodological note on qualitative research

Interpretation and limitations of qualitative research

When carrying out qualitative research, the focus is not to quantify but to explore underlying views, experiences, and motivations. Qualitative samples are purposive and quota-driven in nature; they are designed to achieve specific outcomes such as including a good range of different customer groups relevant to the research. Consequently, they cannot be used to identify proportions of populations holding stated views. For these methodological reasons, it is not appropriate to present qualitative findings in terms of the numbers of participants expressing certain views. The responses should not be seen as representative of the universe but can be seen as indicative. The findings in this report are therefore described in qualitative terms. To give an indication of the strength of a finding from the interviews conducted we have used terms such as ‘a small number’, ‘some’, ‘many’, and ‘most’.

Participant recall

All information reported is based on interview participants’ recall and interpretation, and these are not always precise. Sometimes there is uncertainty because time has passed since the development work they describe was carried out, or in other cases, participants may simply be unsure about key facts or events they convey. Their answers reflect their best and subjective understanding of their experiences. We asked participants to prepare for the interviews to help encourage them to reflect on their view and behaviours, however, it was not part of the research objectives to fact-check, for example, to review any specific project budgets or development outputs. It is a finding in itself that some facts are not retained or possible to summarise easily, and this can relate to the complexity of some decision-making or a lack of priority or perceived need to know that information. We comment on this where relevant in the report.

Individual expert perspective

Participants in this research have provided their personal views in their role as web developers making decisions on development projects in their current role. Each response should be interpreted as being from an individual web developer rather than as an official collective submission from their organisation. Respondents also draw on their wider expertise and knowledge, often from previous roles that they have held, and from their personal experiences outside work as end-users themselves of web apps, websites and mobile devices.

Snapshot in time in a fast-changing market

A key feature of web development is that the pace of change is fast. Participants often add caveats when expressing their views, saying they are based on the last time they developed a particular function or type of web app or website. Respondents sometimes make clear that they may not have an up-to date view of a particular browser feature, tool or framework given updates and changes can happen month-to-month or even weekly.

Our research covers two time periods. The follow-up interviews in 2024 were carried out to sense check that the findings from 2023 were still valid given the fast-moving market. The findings from the first stage of research were echoed in the follow-up interviews, except for the growing use and influence of AI, so the first stage findings are still valid.

Anonymity

Individuals who have taken part have done so on the basis that they will remain anonymous, and their personal details and the company they work for will not be disclosed. This is standard practice in social and market research (and an industry requirement of the Market Research Society Code of Conduct). It aims to encourage open and honest answers from those who take part without fear of any repercussions.

Verbatim quotes are annotated with the individual's role as a front-end or full-stack developer, their type of employment and seniority, to indicate the type of respondent but preserve their anonymity.

3. Main findings

3.1. Web developer attitudes and behaviours

Before looking at the main findings, it is helpful to understand the key attitudes and behaviours of those interviewed. An overview of key points is provided below, with detailed references where relevant throughout the main body of the report.

3.1.1. Personas

Web developers interviewed in the research tended to fall into two groups or 'personas': 'Innovators', who worked more at the cutting edge of development, with deeper technical understanding and opinions about the ecosystems in which they develop; and 'Nine-to-fivers', who were more focused on ensuring delivery and often developed less complex projects.

Web developers reported a constantly changing web environment, where devices, operating systems, browsers, and developer tools evolve fast. Keeping up with this change and adapting how they work is core to any developer's role, but there are differences in mindset.

Innovators	Nine-to-fivers
More likely to:	**More likely to:**
Work on more complex/diverse development projects.	Work on less complex web app and web development projects.
Enjoy problem solving.	Enjoy project delivery and speed.
Keep up and experiment with technical innovation e.g. new approaches, AI.	Use frameworks and tools that help make delivery easier.

The common traits of these two groups and the implications of these are as follows:

Innovators: Around a third of participants fell into this group. They were more likely to be full-stack developers and have a wider range of project and technical experience and to have worked in the industry for longer than Nine-to-fivers. However, being in this subgroup was also, to an extent, about attitude and behaviours, not just job role or level of seniority. They were more likely to develop in Firefox or Brave, or to be flexible about how they approached development. Innovators were more likely to express views on the wider development ecosystems and see benefits and drawbacks to the current environment. They were also likely to be experimenting more with AI.

> *"I am aware that the majority of the world runs on three key browsers for now, Apple Safari, Mozilla Firefox, Google's Chrome browser. And then there's Internet Explorer, now Edge. Yes, inherently for three of those of four I need to absolutely smash it. But it's part of my passion, I'm always reading on what the latest kind of web technologies are… I will always see how the browser wars are, how they're all doing against each other."*
>
> **Full-stack, Freelance, Senior, Innovator**

Nine-to-fivers: Around two-thirds of participants fell into this group. They were more likely to be front-end developers or develop simpler web apps or websites (e.g. the equivalent of online 'brochures' where there is no or low requirement for end-user interactions) and so in turn seemed less likely to come across issues or constraints in functionality. Their goal was a great web app or website that delivers to the client brief on time and on budget, and their client brief(s) typically did not require a complex or cutting-edge solution. Some had a strong preference and/or mostly developed for one main ecosystem e.g. Apple. They were less likely to be aware of underlying technical concepts, and it appeared that it was not necessary to have detailed knowledge for their work. They were more likely to accept the status quo in regards to the current mobile ecosystems. They knew AI was an important emerging influence on web development, but were less likely to be using it yet.

> *"At the moment, browsers are pretty good, but our system (long-term customer's large enterprise piece of software) is a fairly vanilla system, so we're not really asking a huge amount, we're not really pushing the edges of what the devices can do."*
>
> **Full-stack, Freelance, Senior, Nine-to-fiver**

Please note that the proportions of Innovators and Nine-to-fivers among those interviewed might not be representative of the proportions in the wider web developer population.

3.1.2. Development process

Web developers described the process of building web apps and websites for mobile as building once then checking for compatibility across browsers, operating systems and devices. Ongoing web app and website maintenance also formed a significant part of their work.

At the start of the interview, developers were asked about their approach to optimising web apps and websites to ensure compatibility with specific browsers and browser engines, including which browsers they prioritise optimising for.

The main perceived benefit of building a web app or website as opposed to a native app was that developers only had to build once, rather than build separate apps in separate code for different Apple and Android devices and ecosystems.

An overview of the typical development process participants described is below, with the detailed decisions along the way being driven primarily by the initial client brief and developer preferences.

Establish the brief	Develop	Make compatible	Launch and maintain
Understand key client* requirements i.e. the web app/website purpose, and browsers and devices used by the client* and their target customers.**	Develop initially for browser/device which are developer's preferred choice and/or best fit the client brief.	Make compatible across other main browsers/devices relevant to the client brief, and test.	Launch and maintain.

Establish the brief: Participants stated that the client brief is key to decision-making about how a web app or website is developed and optimised. At the briefing stage, they aimed to establish the objectives and features needed e.g. a basic online 'brochure' for a brand or a more complex site to support e-commerce. Another important aspect to understand was who the users would be, and what devices and browsers they favour or use most. Users include:

- ***The client –** meaning the internal/external party commissioning the web app/website who may help test and sign-off the development, and those who will manage the web app/website on an ongoing basis e.g. those updating content via the Content Management System (CMS)
- ****Target customer or end-user –** meaning the target audience for the web app/website e.g. client's customers, or client's own staff if the web app/website is for internal use.

Develop: Participants tended to develop on a desktop computer for their 'browser of choice' as a default, especially where the web app or website was for a broad audience. For most participants the main browser they developed for was Chrome – driven by its share of web traffic, familiarity, ease of use and the quality of the developer tools – but a few favoured Safari, Firefox or Brave. The functionality of the browsers played a role too, but other factors mentioned were bigger drivers of browser choice. There were cases where the client brief rather than the developer's preferences dictated the development approach, as the brief was biased towards or only required support for a specific device or browser type. A couple of examples mentioned included a web app for an upmarket fashion event where the audience mainly used iPhones and a B2B web app/website for field staff who only used Chrome and Android tablets.

Make compatible: Developers tended to make web apps and websites compatible with browsers with the biggest market share, namely Chrome, Safari, sometimes Firefox, Brave or Edge; but rarely Internet Explorer now as its use has significantly declined. In some cases, if the web app or website was relatively simple, or the client or end-users only used certain devices or browsers, then the level of work undertaken to test compatibility would be reduced. A few in the follow-up interviews mentioned using AI APIs/tools to help write code and ensure compatibility, which saved them time. Designing and testing compatibility can range from more basic checks, such as manual checks on a colleague's phone or using simulators such as BrowserStack, through to automated testing across multiple physical devices by specialist third parties. A minority of developers mentioned projects with detailed testing, e.g. by QA team/specialist agency.

Launch and maintain: Many developers mentioned ongoing updates and fixes related to the changing web environment as a significant part of their workload. These are needed to ensure any web app or website still functions efficiently following operating system updates, browser updates, updates to APIs/plug ins, introduction of new devices with new screen sizes, and the emergence of new security threats. Ongoing management can also be needed to fix poor site management by others e.g. the client uploading un-optimised images that slow a site down.

In addition to this contextual overview of the development process, more detail on the drivers of choice and issues encountered during development are discussed in section 3.3 onwards.

> *"Yeah, always Chrome. We focus on Chrome first, Firefox second, Safari Third. That's our order for getting things to work. If we've got it to work on Chrome and Firefox, we're pretty happy. If we've got to Safari, (it means) the team's got spare time on its hands!"*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

3.2. Industry trends and challenges

3.2.1. Trends

Key recent trends mentioned by web developers included a move towards 'mobile first' development and greater use of web apps and PWAs (progressive web apps) in the last five years and the uptake of AI in the last year. The other key trend mentioned on top of this was the increased ease of web app and website development over the last few years.

Participants were asked to share the key trends in the last 5 years in their work on developing and optimising web apps and websites for mobile devices.

An overview of the key trends perceived by web developers is as follows:

Mobile first: Participants stated that, given website traffic is increasing via mobile devices, optimising web apps and websites for mobile devices is often more, or as important, as optimising web apps and websites for desktop. This means designing for smaller screens, which in turn can mean an increased need for: simple UX (user experience), e.g. clean design and layout, minimal content/text, minimal functionality; good responsive design – scaling of text/images for small screens; fast loading e.g. optimising data access and use, compression of images/videos/animations (although 5G is helping with load times); and a trend for single page apps (see web apps below).

> *"Every single website I've created in my whole career, I've had to make sure that it is responsive on all screen sizes. The company (client) I'm working for now, they have a mobile first approach, so we developed for mobile first and then moved up to other sizes. So the main focus is mobile, because most people are using mobile to access the web these days."*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

Web apps/PWAs (progressive web apps): Many mentioned a shift away from native apps and towards web apps/PWAs as the latter now offer some key benefits: good user experience; similar functionality to native apps (or at least sufficient for many kinds of uses); easier development than native apps, as developers can 'build once and it works for everything', using one language across devices rather needing extra skillsets/languages for iOS and Android or tools like React Native; easier development as a first option or for a tight deadline or budget; and instant updates, e.g. no need to wait for Apple to approve updates as is the case for a native app.

> *"Originally, a lot of clients wanted a downloadable app. Now, a lot of customers are moving away from that to a more web app-based system. So we tried to embrace both. So we have a web app solution that we can do with the sort of tooling that we have. We're able to use the logic that we've created for mobile operation, then just have a kind of thin veneer of either a web app or a native app."*
>
> **Full-stack, Freelance, Senior, Nine-to-fiver**

AI: AI was an emerging issue in the main fieldwork in 2023, and the follow-up interviews in 2024 showed this is now having an impact on the speed at which developers can achieve some tasks including: writing code; solving coding queries (where most who have used it in this way feel it is more up-to-date and reliable than developer forums); cleaning up/simplifying code; and adding comments to code so developers can interpret and amend others' coding. AI was not seen as infallible and manual oversight and checks were still needed. More change and opportunities related to AI were anticipated, and this is discussed in more detail in section 3.6 on the future outlook.

Ease of developing web apps/websites for mobile: Perhaps of most relevance for this research project was the repeated mention by respondents that developing web apps and websites for mobile devices is much easier than it was 5 years ago. This is driven by various factors, including evolution in the tools and frameworks available and standardisation across browsers and ecosystems, with AI also beginning to help some too. Detail on the key drivers is listed in the grey box.

Overview of key drivers making web app/website development for mobile devices easier

Frameworks and tools: Frameworks and tools for development are mentioned by many developers, with examples including React, Angular, Node, Vue, Bootstrap and Solid. These are heavily used and highly valued as they are:

- Component based, so you can re-use modules, and different teams/developers can also work on modules separately at the same time; and
- A significant help with compatibility across browsers and devices, so that fewer manual changes are required.

Web apps/PWAs: The increasing use of web apps is making it easier and quicker to deploy and manage apps compared to native apps, as you build once (rather than for each specific operating system) and also can make instant updates as needed.

Standardisation of browsers: Browsers are perceived to have improved functionality for both users and developers, and are becoming more standardised, so fewer compatibility issues arise. In particular, the decline in the use of Internet Explorer (IE) means developers no longer have the challenge of making web apps and websites work with IE unless there is still some legacy use by the target end-users or client. IE was seen by participants to have many issues including poor developer tools, limited functionality and bugs.

AI: The emergence of AI is bringing speed to some tasks as mentioned above, for example, in terms of writing, editing, simplifying and annotating code.

Resources and advice: Participants cited a variety of resources that they valued where they could trouble-shoot and find advice or solutions that made their work easier. Examples mentioned were AI tools, online communities like Stack Overflow and code repositories like GitHub.

> *"Now is the easiest time to be a developer. Tech tools have matured so much. There's React, Vue and frameworks."*
>
> **Front end, In-house, Junior, Innovator**

3.2.2. Challenges

Though developing web apps and websites for mobile devices is getting easier, there are still some common day-to-day technical challenges. Key challenges stem from the existence of multiple mobile screen sizes, multiple versions and updates to ecosystems and tools, and the need to maintain security. Ensuring browser compatibility is a part of this but is not a top-of-mind concern.

Participants were asked during the interviews about the main challenges they felt they faced in their work developing and optimising web apps and websites for mobile devices.

Participants mentioned that it is a constant challenge to keep up-to-date with web technology whilst also delivering their day-to-day work. This can mean, for example, keeping abreast of the latest tools and frameworks or, more recently, with the rapidly changing opportunities potentially on offer though new AI tools.

> *"As a developer, you have to get better at using it (AI), basically, otherwise, you will be out of the game."*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

More specifically, when developing web apps and websites for mobile devices, the following key challenges remain, despite the increasing ease of development compared to a few years ago.

Multiple different mobile screen sizes: Adapting web apps and websites for multiple different screen sizes was the most mentioned challenge. This involves good responsive design, ensuring content and layout is developed with small screens in mind. It is a core need when developing new web apps and websites, but also an issue for existing sites when new screen sizes are released on the market. Developers pointed out there are multiple providers of phones e.g. Apple, Samsung, Google, multiple older models still in use, regular releases of new devices of differing sizes, even mobile phones with two screens e.g. Samsung Galaxy Z Fold, and there are also differing screen resolutions.

> *"Recently there are more and more devices coming out. Screens are bigger and resolutions are different. Even that Samsung that opens into two screens, you have to take that into consideration. When you're building a mobile app, you now have to take into consideration the tablet resolution because screens have got bigger. More devices means more work for us to test it out. Building an app for Android, you would prioritise Google Chrome. Building a web application as a PWA, you would prioritise all of them. You have to check Safari for iOS and Chrome for Android. You don't just have two types of devices like iPhone and Samsung, you also have other devices to consider."*
>
> **Full-stack, Agency, Senior, Innovator**

Multiple system and software updates and versions: Participants often referred to the challenge of ensuring good functionality for new and existing web apps and websites given regular updates within the web environment and continued use of previous legacy versions. This meant developing for multiple versions of operating systems, browsers, search engines, SEO (search engine optimisation) policies, tools and frameworks, APIs, plug-ins, libraries and so on. The types of updates described included new product releases, updates to fix security issues or bugs, or, in a few cases, switches to more up-to-date types of services, such as a different kind of framework with better functionality e.g. a couple of web developers had switched to using Solid as a framework as it was seen as newer and better than React).

> *"People are using old operating systems or older hardware. We have to decide whether to support this or add code to work around different conditions – it's harder to bug fix. When IE was doing things its own way 6-7 years ago it was an issue."*
>
> **Full-stack, Freelance, Senior, Nine-to-fiver**

Maintaining security and privacy: Meeting GDPR and increased privacy and security requirements on mobile, e.g. Apple being strict on 3rd party access to user information. Maintaining protection of security e.g. new threats, potential vulnerabilities from using multiple libraries and tools.

“You hear more and more in the news that there are vulnerabilities and new banks are getting hacked and stuff. It's something that I'm struggling with, I think it’s a constant challenge how I make sure that whatever I'm developing is secure and it's not vulnerable to attack or threat. I mean, I probably could go on and on, but for me, the two biggest ones (challenges) are probably responsiveness for the screen sizes in the market and also the security of whatever it is that I'm developing.”

Full-stack, Freelance, Senior, Innovator

“I guess, because of the nature of technology, something that I build even six months ago, with what is – at that point – the latest version of the language that I'm using, or maybe a particular plug-in that has a set of features for a website. I might have to go back and update it. And in fact, a thing they always have to do anyway is update things in the background because they are constantly updating with new security patches, new features, all that sort of stuff. So I would say that just because I deliver a website today, I still have to do ongoing maintenance on that, and things like regulation changes and things like new technologies coming through means that I need to be on the ball when it comes to making sure that the sites that I have built are still able to continue operating in a world where everything else around is continuously improving and updating itself. So, GDPR was one of the biggest things.”

Full-stack, Freelance, Senior, Innovator

In addition to the above, other challenges were cited by a few participants. These include issues related to developing for mobile devices in general rather than any specific eco-system, such as how to ensure good speed of download and access to data or the challenge of on-going maintenance of websites and web apps once built. Ensuring compatibility across browsers and with iOS and Android ecosystems was mentioned but was not top of mind and seen as part of the wider issue of multiple versions and updates of hardware and software needing to be taken into consideration.

“We’re getting phones that are changing size. It’s a matter of optimising a user’s experience for those different devices and making sure that when a person views it on a device, it matches what our designs say.”

Full-stack, In-house, Senior, Nine-to-fiver

3.3. Browsers and browser engines

3.3.1. Overview of research results

Most research participants pointed to different browsers having distinct benefits and drawbacks when developing for web apps and websites for mobile devices, but rarely talked in terms of finding specific restrictions. Most referred to some need for workarounds, with only a few mentions of feeling limited or being unable to develop what was required for their client.

During the interviews, perceptions of browsers and browser engines were explored in several ways. Participants were asked about key features distinguishing browsers, and if there were any specific issues relating to Chrome or other browsers on Blink, or any relating to Safari or other browsers on WebKit. Interview moderators also prompted participants on whether there were key features that were currently not supported by any of the major browsers or browser engines, or that they chose not to use for any reason.

For most participants the main browser they preferred to develop for was Chrome driven by its share of web traffic, familiarity, ease of use and the quality of the developer tools, but a few favoured Safari, Firefox or Brave.

Respondents were less able to comment on browser engine differences, either through lack of knowledge or experience of any differences or restrictions, or because improving tools and functionality meant that the issues that they were aware of were no longer a barrier.

3.3.2. Browser perceptions and choice

Browser functionality and features are not the main or only driver of browser preferences and choice when developing. The client brief, including the main browsers used by the end-user and the client, and familiarity and developer preference influence browser choice.

Influences on browser perceptions

Developers' perceptions of different browsers are formed from three main sources:

- Experiences when developing.
- The general perceptions and 'received opinion', among their peers, colleagues, and the wider industry together with perceptions based on what they know about the overall brand, especially on topics where they had not had a lot of development experience.
- Personal experiences as an end-user outside of their work, e.g. if someone has 'grown up' using Apple devices they can feel an affinity to that ecosystem.

All these sources combine to form their perceptions. Developers often don't consciously differentiate between their influences when talking about their views and behaviours.

Perceptions of browsers were often described in terms that were synonymous with or interrelated to browser engines or wider Apple and Android mobile ecosystems, with perceptions of Safari being closely related to perceptions of Apple, iOS and WebKit, and perceptions of Chrome being closely related to Google, Chromium and the Android ecosystem.

Developers were generally more able to comment on browsers that they used most frequently during development and so were most familiar, and were less likely to express views on Firefox, Brave or Edge.

Browser choice

Choice of the main browser to prioritise and optimise for first and prioritise was driven by a range of factors, and the features of the browsers were not the main or only influence. Key drivers were:

- **Target audience browser(s):** The end-users' main browser or browsers is one of the strongest influencing factors. This frequently meant web apps or websites needed to work well for Chrome if the target audience for that client brief was a broad consumer base as this is the most used browser across mobile devices. In a few examples the focus was different e.g. developing a web app/website for an event for a fashion audience who mainly have iPhones and use Safari.

> *"I think if you develop your apps or websites for Chrome, then you are kind of 99% of the way there."*
>
> **Full-stack, In-house, Senior, Innovator**

> *"We did the market research. If you're going into store X (high-end fashion store), you are actually much more likely to own an Apple device like the latest iPhone. So that's what we're going for. And a client might come to me and say we have iPads on site. Well, we're working on (and targeting) Apple now."*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

- **Client's browser(s):** The browser used by the client was also a key factor in choice. This meant the client staff who were going to test and sign-off the site and those in their business who may be involved in ongoing maintenance such as uploading content to the CMS (Content Management System). Again, this often led to Chrome being prioritised, as most clients mentioned used Chrome rather than other browsers, in some cases even if they used Apple devices.
- **Familiarity:** Developers often worked in the browser that they habitually used and with the most familiar developer tools, based on what they 'grew up' using, were trained on, or were advised or required to use in their organisation. This meant for example, that although Chrome is the most common browser, some developers used Apple MacBooks and had been trained to develop for Safari and become accustomed to working with it on most projects.

- **Browser functionality:** Developers mentioned using the browser which they viewed as having the best developer tools and that were easiest to use, which was most often perceived to be Chrome, but a few mentioned Safari, Firefox and Brave. They also mentioned preferring to develop for the browser with the best and most frequently updated features, again typically seen as Chrome, but this was less often cited than the developer interface itself.

Other influences on choice did play a role in decision-making in certain instances, such as the need for privacy or security for the end-user (or occasionally for the developer too, where they disagreed in principle with Google tracking online behaviour) or the need for repeat use. These were more likely to affect overall approach e.g. whether to develop a native or web app, rather than browser choice, but could lead to a choice to develop in a browser that more easily enabled adblocking or prevented tracking, e.g. Firefox.

Participants reported a high degree of satisfaction with the main browsers they used, and few expressed a desire to change their development approach unless a significantly better browser, tool or framework emerged.

Subgroup differences

For Innovators, development experiences tended to be broader and/or influences came from the wider industry. They were more able to describe specific examples to support their views and choices. A few also preferred working in less commonly used browsers like Firefox and Brave.

> *"I use Brave as it uses less resource than Chrome, and I can block ads and cookies easily while still getting access to all the Chromium web developer tools."*
>
> **Front-end, In-house, Junior, Innovator**

For Nine-to-fivers, development experience tended to be narrower and/or influenced by immediate colleagues or company guidelines, or the direct need to deliver their current projects to specification and on time. They were more likely to stick to set approaches, and sometimes talked in generalities or were uncertain about some of their views and choices.

> *"I use a MacBook to develop on as this is what I was given at work. I like the way commands are written on Mac. I'm comfortable with it and it's what I've learnt (to develop) on."*
>
> **Full-stack, In-house, Junior, Nine-to-fiver**

Individual browsers

The main benefits perceived and any challenges encountered with developing for each browser are detailed below. As mentioned, these were often described in terms that were synonymous with or interrelated to browser engines or wider Apple and Android ecosystems.

Chrome – Best developer tools and most end-users

Preferred by most developers

Strengths

- Has the biggest overall market share of traffic and is used on Apple and Android devices – this is known and reinforced through analytics, word of mouth and industry news – it seems almost a ‘default’ browser.
- Used for developing for both Apple and Android ecosystems.
- Felt by many to have the best developer tools e.g. Google Lighthouse, Playwright.
- First to launch and try out new features e.g. new APIs for animation.
- Has a wide range of commonly used associated tools e.g. Gmail, SEO, analytics and ad tools, that enable good design and management of web apps and websites and effective marketing.
- Good for extensions and strong APIs in browser – e.g. user location, different audio output options.
- Best to develop for – if it works for Chrome, it is felt it will work ‘anywhere’ as it is widely-used in the market, across both Apple and Android ecosystems, and has similar underlying technology (standards/browser engines) to other browsers such as Firefox and Edge.
- Has become familiar and is high performing, so there are few reasons to switch away.

Weaknesses

- Few weaknesses – a few mentioned that it is not as easy to block ads/avoid tracking (compared to Firefox and Brave) – to an extent, personal preferences as end-users were influencing preferences for development.

“I prefer browser-wise working on Chrome. Mainly because while developing, it has quite helpful developer tools that will help me diagnose problems and be able to view and see things and elements on the page. […] I've always sort of levitated towards Google, as I've always been a user of Google software and devices like my phone, Android instead of iPhones. So I've always lent towards Google. So it’s natural when I use a browser that I use Chrome, and then, Firefox. It's quite similar in terms of developer tools, so I think that's probably why I use that secondarily.”

Full-stack, Agency, Junior, Nine-to-fiver

“Google are keen to always push new leading-edge features. And then everyone else kind of agrees, we're going to do that for Firefox. Then Apple a year later. Okay, we’re going to have to do it now because everyone else has got it.”

Front-end, In-house, Junior, Innovator

Safari – Best for the Apple ecosystem

Preferred by those who use Apple devices to develop

Strengths

- Most commonly used browser on Apple devices.
- Best for development if you need to make use of Apple hardware/functionality.
- Preferred by some developing on MacBook.
- A few felt it is the best to develop in, e.g. efficient to work with once you learn the commands/ecosystem.
- Catching up with Chrome in terms of functionality.
- Some saw it as better/easier for responsive design.
- Seen as best by some to develop for – if something works on Safari/Apple ecosystem, it is felt by some that it will work anywhere as Safari/iOS has high standards, so if a web app or website has been optimised for this ecosystem, it is less likely to be insecure/unstable.
- Seen as having high levels of security and stability as a browser (and the wider Apple ecosystem).
- For those who prefer it, it has become familiar and is high performing, so there are few reasons to switch away.

Weaknesses

- Slower to introduce new features (by a few months or a year).
- Use of different formats e.g. for video.
- Cannot use third-party extensions.

"I would say my primary would be IOS or Safari, so I would typically develop on or for Apple. That's just my own preference in terms how I like to work. When it comes to testing what I've built, I will test it in more than just the Apple ecosystem, so I will test it on Firefox, Chrome. I will use the majority of the mainstream browsers just to double check. And what I'll typically find is that something that looks fine in one does not look correct or does not look as it should do in some of the other ones. So in a similar vein to I can't afford to be ignorant. I feel that the browser ecosystem is also something I can't afford to be ignorant around, because they all work in a slightly different way. When I did my degree at university, it was a very design heavy degree, and at that point, Apple was always considered. The caveat is I would say that everyone has caught up and you can use any device for anything, really. But at the time, there was certainly more of a focus that Apple devices were seen as the ones for design purposes."

Full-stack, Freelance, Senior, Innovator

Firefox – Fading in use but a few still favour

Preferred by a few who tend to be Innovators

Strengths

- Perceived as independent of the main players in technology – Google and Apple – so is valued on this principle by some, who tend to be Innovators.
- Felt to offer good functionality e.g. can install extensions as is possible on Chrome but not on Safari.
- Seen to offer good privacy and security.
- A few felt Firefox had good tools for developers e.g. for simulating how web apps and websites look on different mobile devices.
- Is fast-loading which can bring efficiency when developing.

Weaknesses

- Declining number of end-users and client users so low priority browser.

"Chrome is widely accepted by developers, so it has built its reputation, which is its big advantage. But on the other hand, Firefox has got a lot of libraries and browser extensions that make it friendly for developers."

Full-stack, Agency, Mid-level, Nine-to-fiver

Brave – Seen to offer greater privacy, but seldom used

Preferred by a one person in our reseach who is an Innovator

Strengths

- Perceived as independent of the main players in technology – Google and Apple – so is valued on this principle by some, who tend to be Innovators.
- One person said it had good functionality (though not as good as Chrome).
- Seen to offer good privacy and security, e.g. built in ad-blockers, Google analytics is blocked.
- One person felt it had the convenience of being based on Blink, but without Google tracking.
- One person felt it used less resource than Chrome when developing.

Weaknesses

- Few end-users and client users so low priority.

"With Brave you can get the convenience of Chrome under the hood but without Google tracking you. I don't like the idea that with Google 'you are the product' – you can't compete against the big players, but it doesn't bother me."

Front-end, In-house, Junior, Innovator

Edge – A significant improvement on Internet Explorer

Not preferred by any developer in the research sample

Strengths

- A few mentioned that it is easier to ensure compatibility than for IE as it is now based on the same browser engine as Chrome.

Weaknesses

- No reason (yet) to use for developing given the strength of other browser developing tools – no-one in our research sample prioritised using Edge as the developer tools are seen as clunky and poor (but a few felt this may change given it is now based on Blink and there are new AI tools for developers).

3.3.3. Browser engine perceptions

Few participants expressed detailed opinions about the differences between browser engines. A minority said that WebKit had some specific limitations when developing web apps and websites for mobile devices.

Detailed points of view about browser engines were not widespread – participants were more likely to refer to generalities about browsers or mobile ecosystems.

The few comments made tended to come from Innovators. Nine-to-fivers (who were more likely to be front-end, junior developers or those designing simple websites) sometimes did not appear to recognise the name of the browser engine(s) they were relying on in their work when mentioned by their moderator during the interview, were uncertain about which browsers ran on which browser engines, or had no opinion about them. As such, it did not seem a priority or necessary to understand what was 'under the hood' for many developers interviewed to perform their role effectively.

"I'm not aware of any rules of which engines can be on Android because we're just targeting the biggest browsers, you know? Probably the work we do covers 98-99% of browsers."

Front-end, Agency, Mid-level, Nine-to-fiver

"I'm not sure I know to be honest. Chrome on Android and iOS is pretty much the same."

Full-stack, In-house, Senior, Nine-to-fiver

"Chrome uses Chromium, as far as I'm aware, back-end side of the browser. And Firefox is also built on that. [NB. This latter statement is not, in fact, the case]."

Full-stack, In-house, Senior, Nine-to-fiver

WebKit

Known for 'quirks' and/or some restrictions

Strengths

- Benefits of WebKit are often seen as synonymous with those described for Safari in the previous section.

Weaknesses

- Some mentions of slower introduction of new features, use of different formats and lack of ability to use extensions whether working in Safari or Chrome. These lead to the need for workarounds and a few restrictions.

Blink (mainly referred to as Chromium by participants)

Very few perceptions of limitations except lack of privacy for users

Strengths

- Benefits of Chromium are often seen as synonymous with those described for Chrome in the previous section.
- Some stated that Chrome, Brave and Edge are all now based on Chromium, and where known, this was often seen as a benefit rather than a drawback. It makes developing easier as it has brought standardisation and raised levels of functionality to more browsers.

Weaknesses

- A few mentioned limitations with privacy for users, as Google can access so much personal data through its range of services and analytics, and it is harder to block ads and trackers.

Gecko

Very few mentions in our sample

Strengths

- Gecko was mentioned by very few participants – when mentioned, views were often synonymous with general views on Firefox.
- There was one mention that having Firefox with Gecko provides a way to preserve user privacy.

Weaknesses

- Few weaknesses were mentioned during the interviews – there were no overt mentions of specific drawbacks of Firefox on Gecko compared to Firefox on WebKit, however, a few stated that using Firefox is similar to using Chrome on Blink and implied that this made it easy and familiar.

3.3.4. Examples of browser and browser engine issues and restrictions

The number of browser issues encountered during development and testing of web apps and website for mobile devices is seen to be reducing. Participants tended to describe browsers and browser engines as having benefits and drawbacks rather than reporting significant issues. However, a couple mentioned specific restrictions.

Participants reported a high degree of functionality in the main browsers they developed for, and in many cases did not describe any significant issues or restrictions to their work. There was an acknowledgement by some of these interviewees that there may be some browser-related issues, but they could not point to anything specific.

"I can't think of anything you can't do on a particular browser."

Full-stack, In-house, Mid-level, Nine-to-fiver

"There's no functionality we can't offer on one that we can offer on the other. So, in that respect we're pretty consistent across (browser) platforms."

Full-stack, Freelance, Senior, Nine-to-fiver

"I don't know – there probably are (some things you can't do on different browsers). We try and use things that are supported across most browsers."

Full-stack, Agency, Junior, Nine-to-fiver

Some respondents cited issues, but said these were past rather than current restrictions, as there was a sense that barriers across browser engines and mobile ecosystems in general were reducing.

> *"These days it's mostly about just the layout. Maybe 4 or 5 years ago, you'd have features that came on Chrome but not on Firefox so you maybe had to use older features to ensure nothing would break. But since then, I don't think most developers need to do this because it mostly works on all browsers. 5 years ago, they may have had to make compromises but not really anymore."*
>
> **Full-stack, In-house, Senior, Innovator**

> *"To share within apps used to be very difficult on iOS and Android. If I remember, this was back in the day, this is a long time ago. I'm trying to remember what happened. I do remember there was something specific that stopped me from putting it live on iOS and Android."*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

> *"On iOS you use Safari which is the main browser. On Android phones, you've got just Chrome as standard. I think they are supported pretty well on most things. Previously, there were some features that you'd have a bit of trouble with, for example, on mobile scroll and hover states and mouse events, they would sometimes not work, particularly in Safari. You'd see it being a bit more troublesome. I think Apple are generally a bit more finicky, shall we say. But, nowadays, the level of support is pretty high on both. I think now you get new features, new animation libraries and stuff like that, where they may be supported on one and not the other initially, but as soon as it becomes something that's quite widely used, then they will generally try and get it supported on both (iOS and Android phones)"*
>
> **Full-stack, Agency, Junior, Nine-to-fiver**

A few participants talked about Safari (and WebKit) not fully supporting progressive web apps, although there was a sense by one person that this situation was improving:

> *"Apple was actually holding back the user experience on progressive web apps – the way back and push notifications to your phone. Basically, a bunch of features that you would get on a native application, Apple didn't want to support in Safari, (as then) the Safari browser could compete with the app store, and they have a monopoly on the app store.[...] I get it from a business viewpoint. They want to kind of hold things back for their own financial gain. Eventually, you know, they have to give in and they have recently given in, and it's going to threaten their business model. But I think it's fine. So, you know, Safari is also offering the same kind of features to create a progressive web app in the Safari browser. It's either already out or it's coming."*
>
> **Full-stack, Agency, Senior, Innovator**

> *"It (Safari) doesn't really support progressive web apps like all the other platforms do. So if you wanted to create something that feels like a native app, that's just like: Hey, here's a button to install this app on the web, and then it runs and feels like a like a native app – that doesn't exist on Safari and it's been around in Chrome for five plus years. It's probably just because that whole standard hasn't taken off. But it's interesting, right? Because I think that seems like something that will make more and more native apps go away. But then you know that both Google and Apple are making a lot of money out of those native apps. So maybe there's a tension there where they're like: Well, we don't want people to move too much over into web apps because then we start to lose our income from our app stores. […] So if you're on an Android phone and you're on an application that identifies itself as a progressive web app, a little pop up comes up saying: Hey, do you want to install this to the home screen? And you just click a button and there it is, and after that you almost don't really know the difference."*
>
> **Full-stack, In-house, Senior, Innovator**

Some participants talked about Chromium (a more commonly used term among participants than Blink) being easier to develop for than WebKit, the latter being seen as harder to work with, sometimes requiring workarounds or in a few cases restricting choice.

A few mentioned that WebKit, unlike Chromium, does not allow third-party extensions or access to information, (It is worth noting that Chrome does not support extensions for Chrome on Android, but a few participants thought that it did. Chrome does support extensions on desktop and other browsers on Android are able to support extensions. Participants did not always specifically differentiate between the desktop browser or the mobile equivalent even though the focus of the interviews was on developing for mobile devices).

> *"I would like Safari to be open to 3rd party plug ins, in the way Chrome is. e.g. 'Last Pass' – enables Chrome to store your passwords. Safari won't allow this, so you have to go out of Safari to recall your password. This closed environment of Safari is both a plus and minus."*
>
> **Front-end, Freelance, Mid-level, Nine-to-fiver**

> *"Mobile wise, I think browsers are all very standard nowadays. For Safari, the one difference there would be mobile app extensions, but it's not really used in any of our stuff. For example, on Android platforms like Firefox and Chrome, you can install extensions such as ad blocks so you don't see any ads on your mobile browser. It's my understanding (that you can't do this on Chrome on iPhones). I believe that specific problem stems from Apple's policies, they lock stuff down."*
>
> **Full-stack, In-house, Mid-level, Innovator**

"It's very easy to install extensions on Chromium-based browsers as opposed to Safari. […] most of the things will be the same on both Safari and Chrome but Chromium, is widely used everywhere, I'm not sure what the current market share is but it's over 60%, and all the other spin off browsers are based off Chromium anyway."

Front-end, In-house, Junior, Innovator

"They (Apple) are really strict on what information they will let third parties collect which recently caused a dispute between them and Facebook, because Facebook wanted to collect more data and Apple just wouldn't let them have it basically. It's just something I followed, but I don't actually do in my work."

Full-stack, In-house, Senior, Nine-to-fiver

"Generally, I think people prefer to just work with Chromium-based browsers. From what I know, a lot of people avoid developing on Safari just because it's not easy to do. It has its quirks when it comes to the development tools that it has integrated. When you click on inspect and, we use, for example, we use Vue JS, which a plug-in that you can install as a Chrome extension or as a Firefox extension. When it comes to Safari, well, it's a completely different process because it doesn't easily allow you to just click on install and then everything works. You have to download some separate things, make a server off online, and it can communicate somehow. So it's more steps, and I think there there's no particular reason behind this, apart from Apple making it harder for exactly no reason."

Front-end, In-house, Junior, Innovator

There were very few overt or specific examples cited of development work or choices being impacted by functionality restrictions, because of browsers in general or differences for the same browser across different browser engines (e.g. Chrome on WebKit leading to more restrictions than Chrome on Blink). There were also very few mentions of restrictions seen for web apps compared to native apps, or with in-app browsing.

A few mentioned that web apps in general have a slightly poorer look and feel than native apps:

"Web apps now can access a lot of the system specific hardware as well. So you have access to the camera, the gyroscope, the GPS, to every system that you could on a native app as well. That allows you to do everything that native apps can do. So the only downside would be that it doesn't look 100% and it doesn't feel maybe as smooth as a native app. I think a lot of developers just realise that it's not worth having a separate code base just for those little benefits of not looking 100% as a native up and not having that flare, I'd say that the native app offers even us on our current company. We don't offer a native app, but we have a web app that works just fine."

Front-end, In-house, Junior, Innovator

A few mentioned that they tended to rule out new features to ensure things worked across different browsers and browser engines, one of whom mentioned issues relating to in app browsing contributing in part to this.

> *"So, the advantage on the Apple side is that whichever browser you're using, you know it's going to render the same because it has to use WebKit under the hood. So, even if WebKit is a bit hampered, you can kind of write your workarounds, and then you will know it will work across the board, whereas with Android you might be running Firefox or Chrome or Brave, and they've all got their own completely different rendering engines. You might be just clicking a link in an email, which will open a little browser inside your email app, which is a different browser or it might be the same browser as the one you normally use, but it depends on how you set your phone up, what email app you're using. So there's a lot more to think about on the Android side of things, so you've just got to test more options. So because Apple devices don't necessarily support all the latest features, and because you don't know what browser they might be using on Android, you tend to focus your development down to the lowest common denominator of standards that have been around for a while that you know will work everywhere. You might put in a library, polyfill some bits, so you don't have to worry, then suddenly in the next version of the browser everything suddenly works perfectly. […] So even though Chrome is giving us fancy new features every week, the easiest thing is don't use them as we know they won't work for everyone."*
>
> **Full-stack, In-house, Senior, Innovator**

3.4. Compatibility with different browsers

3.4.1. Overview of compatibility process

Web developers tended to test the compatibility of their web apps and websites for mobile devices against browsers with the biggest market share, namely Chrome, Safari, sometimes Firefox, Brave or Edge. The level of effort put into ensuring and testing compatibility was determined by the client brief and resource available.

During the interviews, moderators asked web developers about their overall approach to optimising web apps and websites to ensure compatibility with specific browsers and browser engines. Moderators also probed for detail on specific aspects, including the use and role of web standards and how much time and cost was involved in ensuring compatibility.

Participants tended to develop in their 'browser of choice' as a default, especially where the web app or website was for a broad audience. For most participants the main browser they developed for was Chrome driven by its share of web traffic, familiarity, ease of use and the quality of the developer tools, but a few favoured Safari, Firefox and Brave. The functionality of the browsers played a role too, but other factors mentioned were bigger drivers of browser choice.

Web developers tended to make web apps and websites compatible with browsers with the biggest market share, namely Chrome, Safari, sometimes Firefox, Brave or Edge. It was rare that participants took time to ensure compatibility with Internet Explorer now, as its use has significantly declined. It was only considered during build and testing if there were legacy reasons to do so e.g. some of the target market or the client still used this browser. This decline was welcomed, as IE was the browser most often cited as presenting compatibility challenges in the past.

> *“We developed Chrome first because that's what's on our computers, what we're used to. A couple of weeks before we hit the end, we basically looked at websites out there that tell you the most common browsers and their common versions. We picked all those versions up and just opened the website and went through it end to end. We prioritised so that we were hitting 80-90% of the market. We did not care about the last 20%.”*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

A few participants talked about a process of developing and checking across more than one browser and/or screen size as they developed, switching between them to ensure they ironed out issues as they went, essentially, trying to pre-empt too many issues arising during the final stages of compatibility testing.

> *“I build using mainly Google Chrome. But then I flick between Safari and Google, because the responsive side of checking the size is a lot better on Safari but the actual development is a lot better on Google, like inspecting elements and stuff. So, there are certain elements which you would say are better on one than the other.”*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

> *“I often just do quick manual checks as I develop.”*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

> *“Essentially during development, I would just be developing on desktop Safari and then also testing it on Safari iOS (for mobile devices).”*
>
> **Full-stack, Freelance, Senior, Nine-to-fiver**

Designing and testing compatibility can range from more basic checks to detailed checking. Use of simulators such as BrowserStack were mentioned by many of those interviewed.

> *“We will develop and test in Chrome. We just go – if it's working in Chrome, we expect it to work in Safari, Edge and so on. We might use BrowserStack to emulate a phone in the way that you can't with the browser.”*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

A few mentioned projects with detailed testing e.g. by a QA (quality assurance) team or specialist agency.

> *"When it comes to testing, we see if it's the same font size on all the browsers or if the images are the same size in all the browsers. Things that are a bit more in detail, they are tested by the QA engineers."*
>
> **Front-end, In-house, Mid-level, Nine-to-fiver**

In some cases, overall efforts to test compatibility were described as minimal, such as manual checks on a phone belonging to the developer or their colleague. This could be because the web app or website was relatively simple, the client or end-users only used certain devices or browsers, or resource is tight. Developers reported that frameworks can be so reliable that they are confident that most web apps and websites will work across browsers and devices.

> *"If the client has got some analytical data, I can see browsers their customers are using and where the traffic is coming from and I see that that's the place to focus, if possible. If not, then I would just look at the latest Google trends and see the most popular browsers and systems and then test across those devices. Occasionally, a client might say, oh, I use the Opera browser or something, in which case, I'll check in that, but quite often it comes down to time as well. There are so many devices out there and systems that it's impossible, really, to test across all of them. There are tools out there for some stuff, but I tend to just do it myself with browsers on different devices."*
>
> **Full-stack, Freelance, Mid-level, Nine-to-fiver**

> *"I actually test myself. We've got devices laying around, so you can just load it up and see if it works. With iPhone. I'm like, cool, it works. You're just making sure that it doesn't break the containers. The tools (Vue) are so advanced, they make your life easy."*
>
> **Front-end, In-house, Junior, Innovator**

3.4.2. Resource to ensure compatibility

Most participants felt ensuring compatibility across browsers was a small part of their work. Most estimated that it typically took 5-10% of their time, however some estimated the time taken was outside this range, with a few saying it took very little or even a negligible amount, and a few others that it took 20-25% of their time.

Compatibility checking is ‘costed in’ to development projects in the sense that the time and cost of this is factored in during the planning stages, as part of the development of a web app or website, and also as part of ongoing maintenance. It was seen as an accepted activity within the work of a developer.

Time spent on ensuring cross-browser compatibility has declined significantly compared to five to ten years ago because of the use of frameworks, increasing ‘standardisation’ of browsers, and reduced focus on IE. Given this reduction in time needed, ensuring cross-browser compatibility was not seen a significant frustration. As mentioned earlier in section 3.3.2 other issues were more often mentioned as a challenge to developers, including ensuring good display on a wide variety of screens, the changing web environment in general (having multiple versions and updates of operating systems, frameworks, libraries, related apps like search engines and SEO tools, and so on, as well as browsers), and maintaining privacy and security.

Participants’ estimations of the time they spent on ensuring compatibility across browsers ranged from around 1% to 25% of their time, depending on the nature of the client brief or sometimes the resource available in the development team. The most common estimate made by participants is that the proportion of time to ensure compatibility was typically 5-10% of the overall time spent on a development project.

> *“If it’s modern browsers (i.e. not including legacy browsers like IE), then it doesn't take that much effort. It's hard to say what time, you're probably looking like 10%?”*
>
> **Front-end, In-house, Junior, Innovator**

> *“When doing development for mobile stuff, probably 5% (of time ensuring compatibility). No, up that a little bit. If we're saying per year, probably 6-8%?”*
>
> **Full-stack, In-house, Mid-level, Innovator**

> *“I won't say it is a significant proportion, but it is a massive finishing thing to do, at least 5 to 10% of what we do.”*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

> *“These days is it doesn't really take a huge amount of time. Most of it is checking the compatibility rather than making them compatible. […] So this current application, it will hopefully be quite a large application, and we're expecting a lot of users to use it. This will probably require a very significant amount of time in testing. I’d be surprised if it's more than a week, because just so much is happening. For this project, that will probably be 10% of my total time. […] I know developers who spend more time than me. I know developers who spend basically zero time testing.”*
>
> **Full-stack, Freelance, Senior, Innovator**

A few mentioned that compatibility testing took very little time:

> *"This cross-browser compatibility is a concern, of course, but it takes very little time. It's a very small percentage of my work."*
>
> **Full-stack, Freelance, Senior, Nine-to-fiver**

> *"I would say 2-3%, I think, that we face this kind of issue (with compatibility). Say, a multi-select is one of the examples, some styled bookmark links will be another example."*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

> *"It (checking compatibility) doesn't take too much of our time because there's such a monopoly. Once something is working in one browser, we kind of assume it is working. If the client wanted, we could test every browser but because we're an agency, we don't, it's a waste of our time. In a way we could charge, we could be like, 'Oh, we're going to charge you £30,000 or more', but they (the client) would be like 'No, we don't need it'."*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

A very small number mentioned that compatibility testing could take 20-25% of development time:

> *"That's 80% of my work building it for Chrome and then 20% after that is patching it for other things. But my client, if they tell me I'm only ever going to run this on a Pixel tablet, also running Chrome, I can be, well, you just got 20% more development time because now I don't need to patch it for other stuff. I can now spend 20% of my time adding one more feature that you really wanted or doing those final tweaks for you. That's the impact for me."*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

> *"I don't know how much money it (ensuring compatibility) costs because it varies. If it's a small project, we have between 2 days to one week of QA phase. If it is quite a big project, it can run up to 2 to 3 weeks, so it depends project-wise. Most of the time, I would say a quarter (of the time) would be mainly working with different Android or iOS versions making sure that it doesn't break down. […] If it works on Chrome on a desktop, the chances are Chrome on a browser on a mobile device isn't too much different. […] They (the main browsers) are all really similar. It seems to be things like sliders, video images, those things, the more fun stuff you have to make compatible for the different environments, but it's accepted that that's what you have to do rather than it being a massive bind."*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

It was harder for participants to estimate cost but a very small proportion mentioned fees associated with using third party suppliers and tools, ranging from hundreds to thousands per annum:

> *"We did a BrowserStack membership. Annually it's £125."*
>
> **Front-end, In-house, Junior, Nine-to-fiver**

> *"I would say it depends if we've automated the test or not. So let's say we're introducing a new feature. So we have the ability we have access to the ability to run that on multiple live devices, automatically, so we can make a change and then run it again. We'll run exactly the same operation over, say, 12 devices so we can set each of those 12 devices up with a different version of Chrome or a different version of Android and test to see how those are. If we're doing things individually, then that's someone's time to do that. So for each iteration, there will be a multiple of doing that. The services that we use that allow us to test over multiple devices, they're not particularly cheap. It would be much easier if I could just test it on one device here and then pretty much guarantee that that would run across the board. You can test for x number device hours on one or more devices and they'll charge. I think this costs us £4,000 a year for that service."*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

3.4.3. Role of frameworks

Most participants reported use of frameworks when developing web apps and websites for mobile devices, with the re-usable code within frameworks helping to save significant development time.

Many participants talked about the rise in the last five to ten years of the use of frameworks and libraries in development rather than 'building from scratch'. These have become heavily used and are highly valued as they provide developers with components or modules of code that are re-usable. This saved significant time on initial development – their own time, and that of others if there was a wider group working on a project, as different developers in a team could work on different modules separately but simultaneously. Frameworks and libraries also made compatibility checking across browsers and devices simpler, as there were fewer checks needed on re-used code.

Frameworks and libraries for development were mentioned by many developers, with examples including React, Angular, Node, Vue, Bootstrap, WordPress, Beautify and Solid.

"So, when I first started programming, we were still mostly just doing a lot of HTML and JavaScript ourselves. And now I wouldn't dream of starting a project without a framework, by which I mean Angular, Vue and React. […] If we use the example (analogy) of this being a car, I used to machine every part myself, and now I start with something that has essentially a chassis and a whole bunch of places for me to bolt things on. Everyone needs a chassis, essentially, so why was everyone building a chassis to begin with? A framework it basically gives me an engine, a chassis and even the exhaust pipe. Let's take that all straight out of the box and then start plugging stuff on top of that. That is the thing that has happened."

Full-stack, In-house, Mid-level, Nine-to-fiver

"It's called Beautify, which is an open-source library. We don't really need to style it in a different way, we can just use whatever it is, and we don't even have to worry about a lot of stuff. It's just already built. A lot of people are already using it on Safari. It's already done by a lot of other developers and tested. And if there were issues, they would be on the GitHub page for that library. The GitHub page has an issue tracker and people can open it there and can say 'Hey, this doesn't work in here. Or can you change that? This is a bug here', and it's usually pretty transparent when it comes to bugs. So if there was a Safari bug, I'm pretty sure it would be there the GitHub page for the respective library."

Front-end, In-house, Junior, Innovator

"So many things have advanced and matured. You don't have to worry about it (compatibility), I don't need to work in Internet Explorer or write code a different way. And now we've got frameworks like React and they just take care of everything.[…] It's too much luxury nowadays! I would have to worry about this stuff five years ago. Not anymore. And I think this is probably an issue with developers – they just don't care about other browsers, they're just happy as long as it works in Chrome, no one really cares anymore, because it's got the market."

Front-end, In-house, Junior, Innovator

3.4.4. Role of web standards

Greater standardisation played a role in bridging the differences between browsers and browser engines, with a few interviewees making specific reference to resources on web standards to inform their work e.g. around accessibility.

Some participants felt browsers had become more 'standardised' in the last 5-10 years and this meant that compatibility had become less of a task.

When developers interviewed used the term ‘standardised’, they were mainly referring to the fact that functionality was becoming similar across browsers, with poorer performing browsers improving their functionality to catch up with the highest performing browser (mainly perceived to be Chrome), or conversely becoming much less used in the market so no longer needing to be considered by most developers (e.g. IE). Increasing standardisation was most often seen as a benefit rather than a drawback, as it made developers’ work easier, although a few expressed concerns about the future if there is too much consolidation (see section 3.6 on future outlook).

In terms of functional differences:

- Many felt browsers were ‘mostly’ up to date (now fewer use IE which was seen as difficult to develop for).
- A few mentioned the fact that more browsers are now based on Chromium and so have similar functionality.
- A few mentioned a sense that Apple and Android are working more closely together on browser functionality – aligning on features and standards which make web developers’ lives easier.

> *“So in terms of functionality, it's (Safari is) very similar in functionality (to Chrome), if I’m being honest. It's very similar now. They are quite head-to-head.”*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

> *“Safari is Chromium-based to a point as well. They are using kind of the same rendering engine. They just took a slightly different approach to it, but at its core, they are very similar, so most things will be the same. The quirks are an annoyance sometimes. Going back to Chromium, all the other spinoff browsers are based on Chromium. Anyway, so Edge is on Chromium right now, prior to Windows 10, we'd have to test on the old version of Edge which was using a different rendering engine, and it was a mess from Microsoft, to be honest, but they decided to do the right thing and use Chromium as well. So right now, we don't really have that much to test on because it's Chromium and Gecko for Firefox.”*
>
> **Front-end, In-house, Junior, Innovator**

> *“Standardisation across browsers used to be poorer than it is today. So certainly Microsoft would say, yes, this is standard, but the standard has changed, improved, and then it breaks things. So there's been a lot of good work between the big players to try and make standardisation mean standardisation.”*
>
> **Full-stack, Freelance, Senior, Nine-to-fiver**

A few participants mentioned looking up or cross-checking against specific resources and guidelines. These were not front-of-mind with very few spontaneous mentions of this being a part of the development process. Examples of guidelines were:

- A few talked about checking what was feasible using reference tools like canIuse.com.
- A few mentioned checking informal UX 'good practice' conventions on developer communities or fora about common icons or layouts, or learning from big technology players in the market – this was described as learning and utilising useful approaches from others with more resource or experience, and so delivering to a good standard that end-users expect or have grown used to.
- A few mentioned checking against accessibility standards.

> *"Canluse.com tells you if you can use a feature in a browser e.g. grid display. Most are pretty up-to-date."*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

> *"I think most companies try to emulate what Google and Facebook are doing. Because they tend to move the industry forward. If you need a lot more focus on accessibility or anything like that, then they have the manpower to actually accomplish it. Even open-source software like React, provides more and better tools to provide accessibility. If you did not have access to React then you would have to create something by yourself. If you search online, you can find accessibility style guidelines that companies like Facebook, Google, Airbnb have. So most companies try to follow those. There are some others (guidelines), but usually older and not as nice as those by Facebook, Airbnb, Google."*
>
> **Full-stack, In-house, Senior, Innovator**

Standards were often seen as guidelines that technology firms and developers aim to follow rather than rules to adhere to. There are some examples of browsers being perceived as not following web standards or 'best practice', for example, Apple was described as having a different video streaming approach to other browsers and so not being standard, but commonly standards are seen as ideals rather than requirements.

> *"Apple has its own streaming protocols – everyone else's is different. Takes me twice as long when coding for video streaming. So, Safari is not meeting web standards in that sense."*
>
> **Full-stack, In-house, Senior, Innovator**

> *"Web standards are recommendations rather than rules put together by 'working groups' across the globe – by web developers but also other industry professionals. I don't think any big companies are involved, perhaps just sponsoring them? They define recommendations (e.g. which version of HTML or CSS should be used – HTML5/CSS3). Apple and Google then take the recommendations on board to respect the recommendations because web developers have an expectation for how the languages etc. will work so they have to meet those expectations."*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

There was also one mention in the follow up interviews of standards being 'enforced':

> *"There has been a move away from native apps and toward web apps. There was a sense that Apple was 'holding this back' intentionally (i.e. Not giving Safari the right features to make sure progressive web apps worked across browsers) as it would put their app store revenue at risk. They had a monopoly of the app stores so don't want Safari competing with its own revenue. This is changing though – their hand has been forced due to 'court orders', 'European Commission standards'."*
>
> **Front-end, In-house, Junior, Innovator**

3.4.5. Examples of compatibility challenges

Frameworks and libraries saved participants significant time when developing. However, some compatibility issues were reported that required workarounds, for example, to layout, video or animations.

Participants explained that frameworks saved time but still required some effort, for example, to:

- Set up various languages, tools and libraries;
- Ensure good integration; and
- Ensure good security across different tools and libraries.

The web environment and frameworks themselves are also constantly evolving, with updates and new competitors and tools emerging on the market. A few participants referred to having to wait to use a particular framework if it was not yet supported by some browsers, or considering use of other frameworks as a better option if their ease of use and functionality had become better than the framework they were currently using.

> *"Sometimes browsers don't support new frameworks and you need to wait."*
>
> **Full-stack, Freelance, Mid-level, Nine-to-fiver**

"I think a lot of people are starting to use Solid now as a framework. It's a new framework which fixes a lot of things that React had wrong around reactivity. And it borrows concepts from all the other frameworks, all the other old frameworks, and it just makes them a bit better."

Front-end, In-house, Junior, Innovator

Compatibility issues mentioned by participants tended to be minor adjustments that were needed to ensure a web app or website worked well across browsers, rather than major concerns or limitations on developing what the client wanted or the solution the developed wanted to put in place.

These issues related to 'differences' between the browsers, or on few occasions 'bugs' or more complex functionality workarounds that were needed. No single issue was mentioned by more than a few people. Innovators sometimes cited several issues, whereas a few Nine-to-fivers mentioned none.

Examples of compatibility issues given related to:

- Minor adjustments to layout;
- Ensuring more visual, complex features work on different devices e.g. animations, videos, streaming, sliders; and
- Ensuring access to different mobile phone features e.g. camera access.

Some issues were described as generic, e.g. challenges for developing for mobile devices in general, whether to choose to include a feature or not or the ability to achieve overall consistency:

"We have a lot of animation and sliders client requests, and they tend to load on a mobile device a lot slower. It will look absolutely fine on the desktop and it will load within half a second, but if you've got different plug-ins, you're using to load different kinds of animations or maps, and different APIs to pull the data through, or Instagram posts coming in or Google Maps embedded, with different styling, those kinds of things, we have to check – what's the loading time? If load time is goes up, we will ask the designer or the client if they want to move some things away from this page because there are too many. So, generally when you're optimising for mobile, I make sure that nothing breaks from a responsive point of view. Everything looks right. When you get a design, most of the times, it has got dummy data, just a few posts, or just less text. But you have to think, what if that will increase in the real world? When they start adding a lot more content, then what will it look like, how will it affect the loading time? How will it affect the styles, how it looks?"

Front-end, Agency, Mid-level, Nine-to-fiver

“Generally, the biggest thing (issue) is consistency – rendering consistency. Ideally, the thing is, I want to write code, and it just looks the same in every browser and functions in every browser. That is probably the most frustrating part that is still left. Typescript and JavaScript have come so far that really what you can do on a website is fairly limitless.”

Full-stack, In-house, Mid-level, Nine-to-fiver

“Not long ago, with async/await, I had to make sure that was enabled across everywhere I was using it. With async/await gives the ability, rather than using a promise, to just to fill a variable. It’s a really nice way to write the code, it takes three lines of messy code that it's hard for someone who doesn't know JavaScript to read and turns into a one line that is obvious. It’s a great feature. But I had to go and check because when I looked at it, there were warnings on it saying not everyone supports this. And I had to make a bit of a choice and go OK, this is cutting edge enough, but I need not use it. Or can I assume by the time this project is mainstream, it's safe. That's a common theme on cutting edge stuff, which is I won't pick up cutting-edge features because they're not compatible with everyone.”

Full-stack, In-house, Mid-level, Nine-to-fiver

As well as general challenges when developing for mobiles, some issues were highlighted specifically around compatibility on Safari:

“Every now and then, you come across a little bug where the browser doesn't behave like I've developed on Chrome and tested on Firefox. It’s usually on Safari. It behaves differently, there's some function that I didn't know about, that doesn't exist on the JavaScript that runs in Safari. It’s usually some small thing, and once somebody’s taught me, I can figure it out and use something else.”

Full-stack, Freelance, Senior, Nine-to-fiver

“iPhone Safari is more distant from Chrome, so I don't want to use it. I'm likely to hit more stuff around permissions. That last thing, in my personal opinion (as an end-user), is a good thing, but as a prototyper it’s not a good thing, so I have conflicting opinions. If I was developing over a year, it wouldn't bother me because those are things that you fix, but I'm a prototyper, I don't have time to try and fight over and over again with permissions. That's what caused issues around getting the video game to run with Safari and Mobile Safari. We’re actually seeing the microphone, we did everything that Apple said that we needed to do, and they still won't let us in. We had to hack and patch our way around that again. Not ideal. So it is a more frustrating experience. When a client comes to me, I need a good reason to go with Apple. I like to go Android as a default.”

Full-stack, In-house, Mid-level, Nine-to-fiver

“I was working on a feature and some of the styles were not compatible with the Safari browser, so we needed to test it. We can't completely simulate mobile behaviour. I'm not sure why that happened, but later I was somehow able to fix that issue. It was a three-level accordion kind of thing with a dropdown of options. When we open the second one, it would open further options. So this is what we find on Android and everywhere else (but not Safari).”

Full-stack, In-house, Mid-level, Nine-to-fiver

Few issues were raised around compatibility with Chrome. This potentially related to the fact that Chrome was most commonly the browser that was developed for first and prioritised during build in terms of functionality. A few mentioned issues around controls of privacy, which focussed on issues for end-users primarily but also affected choices as a developer e.g. blocking information access, tracking or online ads.

“The issues (with privacy on Google): One is ads, two is tracking, and three is just I'm not part of the Google stack. So it doesn't have a load of benefit to me. […] Professionally I use it and two or three of my colleagues also use the same, Firefox. But I would say the front-end developers use Chrome because the in-line just easier. Yeah, there are more tools available from Chrome.”

Full-stack, In-house, Mid-level, Innovator

“Google doesn’t just own Chromium, it owns our lives – Gmail, YouTube, Google maps, apps and everything, Google. They just have the data for everything. It’s not like that data is stored securely, it's not end-to-end encrypted. So you aren’t the only person that can open it, they can see it. Uh, in fact, if you request the data from Google because, you know, with the GDPR laws, uh, they are obliged to give you the data, Uh, that they own, they're going to give it to you in public. URL. Uh, so every file that you store and you have on Google Drive, for example, I could access it from a URL without you sharing it to me. It will be close to impossible to get the URL, but if somehow it leaks because they have it stored on a database, of course, so if that database leaks, you would have access to the personal information that you have on Google Drive or anything like that.”

Front-end, In-house, Junior, Innovator

A few mentioned restrictions rather than workarounds. These could bring frustration, but for many developers the practical business need of delivering on time and budget for their client were cited as the main priorities over and above providing a perfect or cutting-edge solution.

"I need things to work as fast as possible so I pick the easiest route. My choices are path of least resistance, because often are we given 20 days to build something which means that I can't be messing around trying to get it to work on something just because, I'll pick the thing that works. Generally Chrome works the best, it has the least problems."

Full-stack, In-house, Mid-level, Nine-to-fiver

3.5. iOS and Android ecosystems

3.5.1. Overview iOS vs Android ecosystems

Participants tended to see both positives and negatives for each ecosystem. They tended to prefer either the iOS ecosystem or Android ecosystem, often driven by personal familiarity, and this often influenced their views and choices when developing for each of these environments.

This section provides an overview and comparison of iOS and Android ecosystems. Sections 3.5.2 and 3.5.3 cover the perceived strengths and weaknesses of each ecosystem in more detail.

After exploring web developers' perceptions and experiences with browsers and browsers engines in the first half of each interview, in the last half of each interview moderators asked more widely about the iOS ecosystem and the Android ecosystem, and participants' views on any advantages and disadvantages when developing for each environment.

It is worth noting that when participants thought about the 'ecosystem' they would often include many elements in their assessment, including the main browser and browser engine used on that system, the operating system, devices, the relevant app store, and general views of the company (or companies). So comments about the ecosystem inevitably replicated some of those made about browsers and browser engines and were also sometimes quite loose and generic in nature.

In addition, it should be noted that development work is normally done on a desktop using desktop versions of browsers and their associated developer tools. These tools enable developers to simulate how their code will look and behave on mobile devices. Generally participants did not specifically differentiate between the desktop browser or the mobile equivalent when talking about the different ecosystems.

Furthermore, as noted in section 3.3.2, participants' perceptions of different ecosystems are formed from three main sources (personal experience of developing, general industry perceptions of the different brands and their experience of the ecosystem as end-users). All these sources combine to form their perceptions of the ecosystem, though the exact impact of each is difficult to ascertain. Participants often didn't consciously differentiate between their influences when talking about the ecosystems, however it seemed to be that a participant who was mainly developing for, say, Android, would be basing more of their perceptions of the Android ecosystem on personal development experience, whilst their perceptions of the iOS ecosystem would be formed more by general industry perceptions.

Most participants did not see any of the negatives or weaknesses in either ecosystem as barriers that would stop them developing and/or cause major problems to development for either of the ecosystems.

Open and closed ecosystems

Overall, many of the strengths and weaknesses seemed to be seen as a reflection of an ecosystem whose approach was more open (Android) compared to one that was more closed or a 'walled garden' (iOS). Each of these approaches was seen to have both weaknesses and strengths that reflected this overall 'philosophy', and some participants would develop first, or prefer to develop, for the ecosystem that best matched their personal preferences for such systems. For example, some participants liked the more frequent updates and development tools of Google and Android, and were willing to accept that this brought some perceived challenges, such as around web app and website stability and security, that they felt were less prevalent in the iOS ecosystem.

Security and privacy

Security and privacy of web apps and websites for mobile devices were raised relatively commonly as factors distinguishing the different ecosystems, with iOS being seen to offer more security and privacy than the Android ecosystem as it was a less open system. This meant the iOS environment could be more time consuming or challenging to develop for, but was perceived as bringing benefits in terms of the security of the web apps and websites and for the end-user.

Participants would often make comments about the security and privacy of an ecosystem without differentiating between the two elements, or linking performance on one issue to performance on the other. For example, poor security was related to leaks, data breaches and so on, which in turn could be seen as a privacy risk. Therefore the level of performance in one area seemed to be correlated with the other.

That said, a few participants did specifically mention Google in terms of the company's wider privacy policies. A few see Google as having a wide range of services and ability to access, track and analyse users' personal information, which is disliked by a few from an ethical or practical point of view.

Overall, perceptions were again based on a combination of development experiences and industry perceptions – none had experience of hacking, etc either in their personal or work life.

For many, iOS was seen as having superior security and privacy because of its closed 'walled garden' approach as opposed to the more open Android approach. However, security and privacy were rarely mentioned as a reason to not use or feel constrained when using Chrome.

3.5.2. iOS ecosystem

Participants tended to be less familiar with the iOS ecosystem in general, but recognised some strengths.

Overview

Fewer participants were familiar with the iOS ecosystem, and hence Safari and WebKit than with the Android ecosystem. Many noted that they would develop the site in Chrome, then make compatibility checks in Safari and adjust as necessary. A few favoured developing in Safari first, from personal

preference or experience, or when designing specifically for Apple devices (e.g. when most of the users would be expected to be iPhone users).

iOS was seen as a more closed ecosystem overall, with the benefits that can provide, such as security, privacy and stability. Some also saw it as the best end-user experience and a minority saw it as the best developer experience.

However, conversely it was also seen as slower (to improve or add more functionality) and more restrictive than Android, with a minority feeling that Apple policies restricted what they (as developers) could do within the ecosystem – meaning they had less ability to innovate and or it took more time to develop in (as they had to spend time on workarounds to overcome restrictions).

Some felt that Apple was less restrictive now than in the past.

Positive and negative features of the iOS ecosystem

The iOS ecosystem was seen as having six general characteristics that were beneficial. It's worth noting that opinion was sometimes divided on these features; for example, more stringent security was welcomed by some, but could also be seen as adding more work.

1. Stability and regulation

   Participants perceived that Apple had more detailed and stringent guidelines around access and permissions. This was seen as helping to ensure higher levels of quality, consistency, and security.

   Furthermore, limiting Apples devices to one browser engine was seen by some participants as helping to simplify their jobs and save time in general.

> *"There's more control and consistency, but it's also more restrictive. You can't install extensions to Chrome and Firefox because they run on WebKit. But my impression is it's a more secure ecosystem. Apple have high standards even if they are slow."*
>
> **Full-stack, In-house, Mid-level, Innovator**

2. The end-user experience

   Some participants thought that the end-user experience was better on Apple devices than Android. This related to the user interface and it was felt by some that the experience was more consistent, more unified and faster.

> *"The advantages are that you get a more unified experience across Apple devices because it's a common Safari engine. So this ensures that the websites look and work the same on all the handsets.*
>
> *Also, there's tighter control and security protections through the App Store review process."*
>
> **Full-stack, Agency, Mid-level, Innovator**

3. Security and privacy

   Apple tended to be seen as more secure by most participants, whatever hardware, or software they preferred to develop in. In large part this seemed to be driven by the more stringent guidelines and protocols around permissions that were seen to be in place in the Apple ecosystem. This view was also influenced by the perceived “walled garden” nature of the Apple ecosystem in general (e.g. more controls over apps, limiting third party access to data, iOS being seen in general as a more secure operating system).

   It should be noted that for some developers, the tighter security was not always seen positively as it was seen as adding to their workload.

> *“I think Apple has good security; I only have to install anti-virus things very infrequently. But I used to have to do that all the time when I used to use Windows.”*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

> *“I’ve very little experience, I’m more comfortable with Windows not iOS. The perception is it’s more secure, there’s more privacy.”*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

> *“iOS is generally more strict around security and locking down certain features, partly due to Apple positioning iOS as more secure and also the closed-source nature of iOS vs the open-source Android allowing for more manufacturer customisations.”*
>
> **Full-stack, Agency, Senior, Innovator**

> *“It is more locked down and optimised for security purposes which can make some things more difficult for developers.”*
>
> **Full-stack, Agency, Senior, Innovator**

> *“iOS is more secure because of the restrictions to browsers and browser engines.”*
>
> **Full-stack, In-house, Junior, Innovator**

> *“Apple has a stricter approach to permissions and tracking, which is a positive from a user security and privacy perspective.”*
>
> **Full-stack, Agency, Senior, Innovator**

4. Familiarity and habit

   A few participants noted that they were very familiar with Safari due to their previous or current experience working with Macs and the wider Apple ecosystem. This was seen as a positive as it meant that they had experience working with Apple products generally so they knew general commands, and how things worked in that ecosystem. It was therefore natural to tend to favour developing first for the Apple ecosystem.

   A few did note that Safari could take longer to learn to develop in initially, but that once it had been mastered it was then easier and more elegant to develop in than other browsers. A minority also commented that developer tools on Apple had improved in last 12-18 months

> *"Bit harder to learn the commands that work for developing in the Apple ecosystem but once you learn them, it is straightforward to use and nicer to work with."*
>
> **Full-stack, In-house, Junior, Innovator**

> *"I like the way commands are written on Mac, I'm comfortable with it, it's what I've learnt on… I'm not sure what is offered on Android so can't really compare, but I'd assume most of the software I use would be available."*
>
> **Full-stack, Agency, Senior, Nine-to-fiver**

5. Device compatibility

   Most participants noted that when developing within Apple there were very few (if any) compatibility issues between devices. It was mentioned that if a web app or website or worked on one Apple device it would often work on all of them. This helped to save time when compatibility checking for different Apple devices and was a contrast to the Android ecosystem.

6. Technical features

   Some participants felt some technical features made developing easier and more intuitive for iOS and Safari than for Chrome and in the Android ecosystem. For example, it was mentioned that there was less caching on Safari than Chrome, so that it was easier and faster to use when developing.

> *"I enjoy the Apple and iOS design language and user experience, it's really intuitive and feels "human"… I also like the ability to sync data across different Apple devices… and my Mac is great for creative work and development."*
>
> **Front-end, Agency, Junior, Innovator**

Participants also mentioned five characteristics that they saw as negatives or limitations. These are detailed below:

1. Slower release of improvements

   The iOS ecosystem was seen by most participants as much slower than Android to release improvements and new functionality. It was noted that in some cases Apple could be several months or a year 'behind' Android.

   Participants noted that this meant that they had to expend time on finding workarounds or simply put-up with the fact that some functionality would not work in the same way on Safari and iOS as on Chromium-based browsers.

> *"Going back to Apple, they didn't bring support for web apps until I think iOS13, and Android had it years ago. Web apps can now access system-specific features now like camera etc. That allows you to do everything that a native app could do."*
>
> **Front-end, In-house, Junior, Innovator**

> *"Restrictions on browser engines in iOS cause headaches occasionally. I recently had a bug related to the URL bar that took 1.5 hours to fix."*
>
> **Full-stack, Agency, Mid-level, Innovator**

2. Apple protocols and standards

   Some saw Apples protocols and standards as the flip side to the 'stability and regulation' mentioned above, effectively slowing down innovation and improvements.

   Apple was also noted by a minority as having its own 'unique' protocols for certain functionality. This could be seen as just different, or suggested to some that Apple was either not adhering to web standards and/or was slower to adopt them. The negative impact here (as mentioned by a minority of participants) was that they had to spend more time developing workarounds to get the functionality they wanted to work.

   It was also noted that Apple protocols were often stricter than Google and Android.

> *"Video streaming – Apple has its own streaming protocols – everyone else's are different. Takes me twice as long when coding for video streaming. Safari is not meeting web standards in that sense. Bluetooth beacons – Apple has its own standards – it's tricky for developers to work out which is the right way. Multiple Bluetooth beacons is too much of a headache for developers."*
>
> **Full-stack, In-house, Senior, Innovator**

3. Reduced feature sets/lack of ability to customise

   A small number of participants noted that on Apple they were unable to use the full feature set of browsers – this was seen as restricting their flexibility and creativity and also limiting the level of customisation they could apply.

   Some developers noted that they preferred the developer tools that were available on Chrome, which weren't available on Safari.

> *"iOS doesn't let you gather user data unless users opt in, so you have to use external libraries to do that. On Android it's a bit different because apps are not sandboxed like on iOS so it's much easier to access something like Bluetooth, Wi-Fi or location data. It has its cons but also its pros because you don't have to think about multiple things [on Apple], you can just do one thing, and you know it works. On Android, they don't check [the apps] the way Apple does. I think they just more focus on getting all the applications they can get for growth of the Play store."*
>
> **Full-stack, In-house, Senior, Innovator**

> *"Debugging and testing on actual iOS devices is more difficult compared to the Android emulation available in desktop Chrome dev tools. The Safari web inspector involves more steps and is buggy."*
>
> **Front-end, Agency, Junior, Innovator**

4. Limitations regarding layouts/visuals

   Some participants mentioned that they experienced problems or complications in relation to layouts and visuals on Safari. A small number of participants specifically mentioned CSS (Cascading Style Sheets) related issues affecting how HTML (HyperText Markup Language) elements are displayed. This meant that they had to spend time building workarounds to get the visual layout they wanted.

   Examples mentioned included:

   - The fact that Safari requires that the outline property rendered as a rectangle even for elements like avatars that are circles
   - Safari can be months behind Chrome in supporting new CSS features so that some developers have to use customised CSS hacks

> *"When you're testing, there's way more CSS and JS bugs in Safari. Safari won't allow you to use 'Flexbox' which is a really useful tool for layouts on mobile. It makes Safari really clunky for Chrome for making web apps accessible."*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

"iOS updates have impacted how tracking pixels and permissions work, that requires adjustments on the web development side."

Full-stack, Agency, Senior, Innovator

5. Ease of learning

   A small number of participants noted that it was not as easy to learn or get started in Safari as it was in Chrome.

3.5.3. Android ecosystem

The Android ecosystem was widely used by participants, with Chrome often being preferred to Safari as the initial choice for developing web apps and websites for mobile devices.

Overview

A greater number of participants stated they were more familiar with the Android ecosystem than iOS, with Chrome being the default browser for development for many (with the default procedure being to develop in Chrome and then check/ensure compatibility for Safari).

The Android ecosystem was often perceived as being progressive and more 'open' (in terms of adding new features quickly, having more tools and having fewer barriers to entry for developers). Related to this, benefits include perceived frequent updates/improvements and a strong set of developer tools. The Android ecosystem was overall seen as easier to use than iOS.

It's also very widely used by both developers and clients, so there may be some element of 'network effects' impacting here.

However, limitations were mentioned by a minority in relation to security and privacy, as well as perceptions that Chrome can be resource intensive. For a few, device compatibility was also an issue, given the wide range of Android devices on the market, particularly in regards to screen sizes (although it was also mentioned that the impact of this issue had reduced over time because of the use of developer tools and frameworks).

Positive and negative features of the Android ecosystem

The Android ecosystem was seen as having seven distinguishing features that were positive. As with iOS, it's worth noting that opinion was sometimes divided on these features; for example, being able to use more open-source third party tools when developing for Android was welcomed by some, but could also be seen as creating security issues.

1. Familiarity and perceived market share

   Most participants noted that they were very familiar with Chrome, therefore it was often front of mind when any new brief and project arrived. As noted earlier, Chrome was intrinsically associated with the Android ecosystem – and when asked about the ecosystem respondents would often answer with comments relating to Chrome.

This level of familiarity meant that they often had a very good understanding of how to work with it, its features, limitations and the tools available for it. For some, this deep familiarity meant that even if another browser was better for certain functionality, it could still be easier (less cognitive effort) for the developer to use Chrome. Familiarity added to perceptions of ease of use.

The familiarity seemed to be itself driven by Google's perceived high market share, in that it was widely used by end-users and clients. Therefore many participants consciously used Chrome to develop with as a default as this was seen as meeting many use cases and client needs. It also meant that most Android users' needs would be met.

> *"Chrome is more familiar to the end-user – it links up with Microsoft and uses the same short-cuts/commands etc. so it's just it's a bit more intuitive, and the same for a developer when coding you don't need to learn loads of new commands."*
>
> **Full-stack, In-house, Junior, Innovator**

2. Links with the Google ecosystem

   A few users noted that Chrome linked easily to their Google account, which meant that it was easy to access different developer programmes, tools, bookmarks, passwords and so on. It was also easy to access Google Workspace products (such as Gmail, SEO tools, etc) which again added to the ease of use of the ecosystem.

3. Open source

   Many participants mentioned that Android (and Chrome) was more 'open source' than iOS. This perception was driven by a general industry perception, as well as experience developing for the Android ecosystem. Being 'open source' meant there was a greater availability of code that could be built on (produced by a large community), and hence more tool kits, libraries and extensions.

   The open-source nature of Android was seen as producing a number of benefits:

   - There were more tools and extensions available so that they had more flexibility when developing.
   - It was simply easier to use overall than the iOS ecosystem.
   - Some participants felt that that the lack of restrictions meant that it was easier and faster to test web apps and websites than on iOS.

> *"So, I'd say the big pro about Android is it's just really open in that it's not very restrictive. It's quite easy to get things deployed and moved, particularly when you're talking about mobile apps. Uh, It's a lot easier than with IOS. They're a bit more restrictive in what you can and can't do. But on the flip side, with Android being more open, it's a bit more vulnerable to security issues. Um, so with Android, the openness can have a positive and the negative."*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

4. Frequent updates and additions

   This benefit was linked to the open-source nature of the Android ecosystem by some, but was mentioned independently by others.

   Many noted that Google had a reputation for being the first to launch and try out new features and updates, and it was often many months in advance of Apple in this regard. The new additions included both new features and also improvements (stability, performance, etc) – therefore these helped drive perceptions of both innovation and performance improvements. For example, it was noted that Chrome was the first browser to allow development of web apps and user location in browser.

   A minority of participants did note however, that it could be difficult to keep up with the pace of change within the Android ecosystem.

> *"The developer tools and suite are brilliant. I think because it is so vastly backed, with Google, there's endless updates, endless stability updates, endless improvements, and it definitely helps the whole experience in terms of the engineering side."*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

5. A strong set of developer tools

   Many participants stated that Chrome has the best developer tools (including APIs), although some believe the gap between Chrome and other browsers is not as large now as in the past. For many this means that Chrome is easier both to programme in initially and to debug. Linked to this point, it was noted by a few that it was easier to install plug ins and extensions to support development for the Android ecosystem than for the iOS ecosystem.

> *"I do like the Chrome dev tools, especially Recorder and Playwright. It allows you to record how your code will play out on devices. Bit like BrowserStack but within Chrome. I use Google Lighthouse for SEO stuff."*
>
> **Full-stack, In-house, Mid-level, Innovator**

6. Compatibility with other browsers

   Many participants felt that developing in Chrome would help with compatibility with other browsers. A few specifically stated that this was because other browsers could also use the Chromium browser engine in the Android ecosystem. Also, many participants commented that it would only take a small amount of time to ensure that web apps and websites developed on Chrome would work on Safari or other browsers as there was increasing standardisation in general.

> *"You can use desktop Chrome tools to emulate mobile devices for testing which works well."*
>
> **Front-end, Agency, Junior, Innovator**

7. Technical features

   Participants would often mention a specific technical feature that they liked or found useful when developing for the Android ecosystem. The examples shown below were each mentioned by a minority of participants:

   - User data collection (e.g. location) is easy, which helps with developing for a good user experience and for marketing.
   - Integration of push notifications is easier than for Safari.
   - Generally good rendering capability (seen as better than iOS by some).
   - Handles modern CSS frameworks and responsive features better than older browsers.
   - Scroll positions (Android handles scroll position calculations differently than iOS, with Android updating smoothly during scrolls).

> *"Features on Android arrive first. So CSS related features, new APIs for animations. For example, when you go on YouTube on mobile and you click on a video and that video just expands to your full screen. So, that transition right now doesn't exist everywhere. It's a feature that will come to the web soon, I think in a few months. Android will pick it up first, iOS will pick it up after a year, at least, So it's slower to pick up new features. Like with web apps, it took them a few years to allow them on the platform."*
>
> **Front-end, In-house, Junior, Innovator**

> *"Chrome also has a 'Performance Insights' function that I haven't seen in Safari – I only really use this when developing progressive apps as they are more technical."*
>
> **Full-stack, In-house, Junior, Innovator**

> *"Generally there are fewer layout issues compared to iOS and Safari."*
>
> **Front-end, In-house, Junior, Innovator**

Participants also mentioned four characteristics that they saw as negatives or limitations. These are detailed below:

1. Device compatibility

   Some participants mentioned that the number of Android devices (and screen sizes) on the market means that device compatibility is more difficult as there are simply more devices (from more manufacturers).

However, many saw this issue as less important than in the past given the tools and frameworks now available. Wide use of the latest frameworks and libraries meant compatibility was mostly coded in and taken account of already. Regular use of simulators during developing and testing meant a reduction in issues with either Android phones or iPhones.

A few participants stated that they simply design for one size and trusted the frameworks they used to render appropriately. A minority would double check with a small number of physical devices.

> *"With Android, it's a thousand different phones so it's a lot more effort to have everything working on every different device. If you're trying to develop an advanced feature like a camera filter or something these days it's usually with AI, on Android it's mostly going to be for the latest devices but for Apple, it's going to be for all devices."*
>
> **Full-stack, In-house, Senior, Innovator**

> *"There's so much you can do with Android. It's almost like it gets messy very easily. So, I prefer IOS. Android based browsers are not able to convert a lot of elements and icons are dumped on instead of converting a simple line break element. So, stuff like that, depending on the version they're using. A lot of people use older Android phones and versions than on IOS."*
>
> **Front-end, Agency, Mid-level, Nine-to-fiver**

> *"There's more fragmentation in terms of devices and operating systems versions compared to iOS… it can be trickier to develop for and test on the range of Android devices."*
>
> **Full-stack, Agency, Mid-level, Innovator**

> *"Some of the older Android operating system versions don't support the newer OSS features, so you've got to do more debugging work."*
>
> **Front-end, In-house, Senior, Nine-to-fiver**

2. Security and privacy

   There was a general belief amongst participants that Android is less secure than iOS. This was driven a number of factors including:

   - The open source storing of data (mentioned by a minority);
   - The fact that permissions are less strict than those imposed by Apple;
   - The large number of third-party libraries and toolkits available in the Android ecosystem (in the sense that it is more difficult to monitor and police them all);and
   - Their personal experience as an end-user (e.g. by seeing 'fakes' on the Android App store).

As noted above, there were also some concerns around Google's privacy policy and how personal data was shared between various products in the ecosystem.

For most participants, however, issues related to privacy and security were not serious enough to stop them developing in Chrome first. For many, 'worse' security seemed to have been seen as simply part of the trade-off of having a more 'open' policy (with the flexibility and other benefits that that brings). A few who perceived it as an issue, either practically or in principle, chose to develop in Firefox or Brave, though this was very much the exception rather than the rule.

"You are often using a package manager (e.g. PNPM, Yarn) some of which is open source. It's common for packages to become obsolete or have security issues which is an issue as other parts of the code and design are reliant on this. You then have to make a call as to whether or not you remove the package altogether or go in and make tweaks to work around. It is a big challenge and difficult to solve."

Front-end, In-house, Junior, Nine-to-fiver

"On Android it's easier to load custom web browsers or apps not approved by an app store, providing more flexibility but less security. iOS has more of a 'walled garden'."

Front-end, In-house, Junior, Innovator

"[Android is] open, not restrictive, easy to deploy apps, but it's also more vulnerable to security issues. There's not a huge difference between Android and iOS really – it's just personal preference."

Full-stack, Agency, Junior, Nine-to-fiver

"Security is another challenge – when using all the tools (e.g. React, Tailwind etc.) so many different libraries go into each tool so you can never be 100% sure it is secure."

Full-stack, In-house, Senior, Innovator

"iOS won't support everything which adds to the security of the ecosystem whereas Android can support multiple things which means security breaches are more likely."

Full-stack, In-house, Junior, Innovator

"Chrome is much more open, with lots of plug-in options. This does mean it's less secure than Safari as the multiple plug-ins cannot all be scrutinised to the same standard."

Full-stack, In-house, Mid-level, Nine-to-fiver

3. Use of computing resource

   Some noted that Chrome was resource heavy in terms of memory use. This seemed to be more in regards to development rather than the end-user experience. For example, it was noted that Chrome uses a lot of RAM (memory) and this can slow computers down when testing, debugging, etc.

> *"Chrome is a memory/CPU hog so I often work in Safari to save battery life on mobiles."*
>
> **Front-end, In-house, Senior, Innovator**

4. End user experience

   Some noted that the end-user experience on Android devices was not as good as on Apple devices, for example, Apple devices were generally faster.

3.6. Future outlook – AI and other change

3.6.1. AI

AI tools such as Chat GPT were an emerging topic in 2023, and, by 2024, were having a clear influence on the development process among many of those who took part in follow-up interviews. This was a trend participants felt would continue and have a potentially major impact on their work.

As noted in section 2.4, this research project was paused in March 2023 for nearly a year. When re-started, 13 participants were re-interviewed to help understand if anything significant had changed in the intervening 12 months. Overall, the re-convened interviews confirmed that the data collected in 2023 was still valid, and remained relevant to inform the CMA's research objectives, but that AI was starting to influence many participants choices and behaviours.

Most of the re-convened participants saw AI as a tool that was increasingly helping them in their day-to-day role and/or had potential to do so in future. Effectively AI was seen as helping them become more efficient in terms of saving time in general and specifically reducing the time they spent on the more mundane development tasks.

Specific examples of tasks it was used for were:

- To solve small coding problems;
- To write basic code;
- To help optimise code;
- To help with ensuring compatibility; and
- To help with annotating code so other developers could understand or amend it more easily.

The most used AI tool was ChatGPT. A few had already seen more client requests for AI to be built into web apps or websites that they were developing. One example was a client who was using AI to help build more personalised customer journeys for its customers.

Some also believed that AI may impact on their role as developers, given that it would bring efficiencies to the process of developing overall. This was seen as something that could potentially lead to job losses, especially amongst front-end designers.

Linked to this, some business owners and senior staff also felt that they may lose some business to companies that offered templated websites and page builders. The logic here was that with AI, these self-serve products would be even easier for their clients to use (without the need for developer involvement).

> *"I'm probably getting fewer requests for work through, I think, these days mainly because a lot of people are exploring how to build websites themselves with the web builders... I think eventually the art of like, building web sites might sort of, you know, die out a little bit. Where people are just using kind of AI to be able to do what they want to do."*
>
> **Front-end, Agency, Senior, Nine-to-fiver**

> *"It's [AI] going to become more and more prevalent, especially in the way you architect and build applications… So it's this case of getting the knowledge in early because then it's easier to expand upon that knowledge base as opposed to, you know, trying to catch up later."*
>
> **Front-end, Agency, Mid-level, Innovator**

3.6.2. Other current and ongoing trends

Aside from AI, other trends that were mentioned tended to be perceived as incremental or continuing rather than 'game-changing'.

Almost all participants felt that adoption of new tools, libraries, languages and so on would continue, and some felt that the pace of change in this respect would increase. Therefore the pressure on developers to stay up-to-date was seen as continuing.

Specific examples of new tools, etc were:

- New/updated React libraries;
- CSS Flexbox, Grids, etc; and
- New/updated ways of managing content, such as Headless CMS.

A small number of participants also believe that compatibility checking between browsers would continue to get easier, driven by the release of the new tools mentioned above. Some felt that AI would help here as it could be used to help address compatibility issues.

Other technology related trends that were noted included:

- Mobile first continuing to increase in importance;
- Improved page builders/DIY sites;
- Increased importance of load optimisation as online traffic continues to increase;
- Continued move towards cloud solutions; and
- Potential increase in use of MS Edge (given it is now Chromium-based and more up-to-date for developers and its early adoption of AI).

More generally, a few participants believed that privacy may become a more important issue affecting the sector, and hence Brave may become more popular to develop in. A few also mentioned that sustainability and legislation and regulation may also increase in importance.

3.6.3. Competition and choice in the future

Most participants were accepting of the level of choice of browsers and browser engines in the market. Most felt browsers and browser engines were more standardised and this made their work more efficient. Many stated the current browser providers had to innovate to compete, with only a few expressing concerns that dominance or restrictions from Apple or Google were limiting choice in their view.

Towards the end of the interview, participants were directly asked about whether more choice around browsers and browser engines would be beneficial or disadvantageous for them as web developers.

Before exploring their responses to this question it is worth noting that participants were answering from the perspective of their ‘day job’ as developers, and deciding on what makes a market competitive is not something many spent much time thinking about.

Feeding into this, it should also be remembered that many participants, especially Nine-to-fivers were not especially engaged with the industry. Many (i.e. Nine-to-fivers), were more focused on simply getting their immediate job done, rather than any industry implications.

Conversely Innovators were more informed, and they often looked at having more browsers as a trade-off between having more choice on one hand, and business efficiency on the other.

Many participants stated that, as developers, they wanted to increase standardisation to reduce testing and amending as much as possible. The phrase ‘one browser to rule them all’ was said by one participant as a humorous comment in regards to this, but serves to illustrate how many were thinking.

Therefore, having more browsers (or browser engines) was not automatically a good thing in itself, if it meant more coding and testing.

Most recognised that some level of competition between browsers was good as it promoted innovation and ensured the providers did not increase costs or otherwise take advantage of developers and users. Many believed that the current situation was a result of this, as the poor performers (such as IE) had died out, and smaller browsers (such as Brave) were able to innovate and fill gaps left by the larger browsers (in this case, offering better privacy than Chrome). A small minority actively wanted more choice of browsers both to drive innovation and sometimes because they simply did not want a duopoly.

Overall, the majority of participants (both Nine-to-fivers and more senior developers, who were often Innovators) did not want a proliferation of browsers and choice as they felt this would cause more work for them. The majority seemed broadly happy with the status quo as it stood at the time of their interview.

> *"I don't think we need more choice. We should have less choice but with the browsers more optimised. It's better to focus on improving current browsers to be the best they can be, rather than more choice. It makes life easier!*
>
> *But at the same time, I wouldn't want one of the big players to have a monopoly. They have smaller players in the mix now (i.e. Firefox, Brave) and I think that works well. I think it works fine now as long as there are the independent alternatives available."*
>
> **Full-stack, In-house, Senior, Nine-to-fiver**

> *"It [Apple restrictions] does slow browser innovation overall by limiting engine diversity in iOS, but also provides more unified landscape that is easier for developers."*
>
> **Front-end, In-house, Mid-level, Innovator**

> *"The dominance of Chromium could be problematic for the open web, which is why having Mozilla with Gecko still around provides balance and fights for user privacy."*
>
> **Full-stack, In-house, Mid-level, Nine-to-fiver**

3.7. Case studies

Case study 1: Frameworks and web apps reducing challenges with browsers for mobile devices

Role: Full-stack
Type of employment: Agency
Seniority: Mid-level
Persona: Nine-to-fiver

Background

This developer works in a consultancy, his main client is a UK utility company but he also develops web apps and websites for a range of other sectors and smaller companies. He manages a small team of developers as well as doing development himself.

Development process and rationale

He often advises clients to build web apps not native apps, but it depends on their needs. He feels web apps and websites are solutions that suit most problems, and progressive web apps also work offline if that's a requirement. It's also easy to find coders for web apps and websites rather than native apps. The major browsers he considers when developing are Chrome, Safari and Firefox. They all have great support and great developer tools. IE used to be a challenge to develop for, but Edge, its replacement, is fine on mobile devices.

Issues with browsers, browser engines and mobile ecosystems

Differences in browsers are not an issue in his view. His work used to be quite manual, but now everything is more streamlined with HTML5 and frameworks like React making development a lot easier than 10 years ago. Ensuring compatibility with different browsers doesn't take much time and is built into the time and costs of each project.

He can find it a bit tricky in terms of permissions with web apps, but nothing is difficult or limiting. For example, when he is trying to mimic a native app, he needs to make sure the web app can access features e.g. file systems. This is easy on Android, but harder on iOS – with Windows it is even more difficult.

Screen sizes can be an issue, as you need to customise across devices not just platforms. Also, Android evolves frequently compared to other operating systems so it's hard to keep up with all the different versions.

Future outlook and AI

One of his main challenges is keeping up with new technology in general. He goes to conferences, reads newsletters, and looks at RSS feeds and online groups. He has not yet used AI in his work. (NB. interviewed in 2023 only, no follow-up interview).

> *"I don't think anyone is designing specific to browsers these days. […] We have new technologies, new frameworks. Earlier, it was a lot of manual things. Simple things had to be done in the past in a complicated manner, but now technology has streamlined."*

Case study 2: AI reducing development time for mobile devices

Role: Front-end
Type of employment: Agency
Seniority: Mid-level
Persona: Innovator
Background

He has been working in development for five years for a variety of big and small clients, including in property and leisure. He has developed websites and web apps.

Development process and rationale

His agency's approach is mobile first. He is mainly WebKit based. He uses Chrome developer tools, works on a MacBook, and ensures compatibility with Chrome, Safari, Firefox and Edge. He doesn't guarantee web apps or websites will work on IE as IE technology is quite backward. IE needs 100 lines of code when you can do the same in 5-10 lines of code elsewhere, and IE does not support two front-end CSS tools, Grid and Flex, so would take a lot of extra design time and can still break in older versions of IE. He has to make sure all web apps and websites are responsive for different screen sizes, using breakpoints for responsive design, and tests on simulators like BrowserStack. He estimates 25% of developer time is spent on QA (quality assurance checks).

Issues with browsers, browser engines and mobile ecosystems

He uses WordPress tools which mean he doesn't have to worry about a lot of things like device or browser. A lot of compatibility comes 'out of the box'. The main issues he finds for mobile in general are sliders and animations loading slowly, and also different plug-ins, APIs, Google Maps etc – sometimes he has to suggest removing a feature for mobile to improve loading. Most initial designs include dummy data only, but he has to consider the style and load times when there is more data too. There is a Google tool to test load times and BrowserStack too.

He feels Apple requires more adaptations during development (e.g. he had to do a workaround to change the style of buttons on a project). However, Apple presents fewer cache issues (he can remove cache for that particular website, so you can see changes) and once developed, web apps and websites are more stable. Android crashes more (e.g. when uploading files). He also had an issue with video on Android not playing automatically for a client when he had low battery, which he spent 2 hours troubleshooting then coding a fix. He would prefer if Chrome developer tools that help ensure responsiveness were as good as Safari's.

Future outlook and AI

He follows the market. If a new browser became popular, he would adapt for it. The limited number of browsers and browser engines means h**e** currently doe**s**n't have to worry or do too much QA. Since 2023 (as mentioned during his follow-up interview in 2024), he has started using an AI API with his code editor to check how compatible his code is and this saves even more time.

“Within my VS (Visual Studio) Code, my code editor, I use an AI API integrated with it. When you’re writing, you select the code and it is like – how compatible is that? Then you give it a command and instead of spending a few hours writing different compatibilities, it gives you the base code i.e. you need this for Internet Explorer, this for Android devices or different screen sizes. So it cuts down development time. But I’m not using anything (AI related) in the browsers.”

4. Appendix: Fieldwork materials

4.1. Recruitment screener

103723 CMA Web Developers Research

Screener

Introduction

Good morning/afternoon/evening. My name is _____, and I work for Roots Research. We have been commissioned by Jigsaw Research to organise some research on behalf of their client, the Competition and Markets Authority (the CMA). The CMA is the UK's principal competition and consumer protection body responsible for promoting competitive markets and tackling unfair behaviour.

We are recruiting Web Developers to take part in an important research project which aims to understand the experiences of Web Developers when developing websites/web apps for mobile. The research will explore Web Developers' experiences of working with the main browsers and browser engines in mobile.

The study is for research purposes only and is not a solicitation for business now or later. Anything you say during the research exercise will remain confidential and you will not be individually identifiable in the findings we share with the CMA.

I'd like to ask you some questions to determine if you qualify for this particular study. The answers that you give to me today will be shared with **Roots Research** and **Jigsaw Research.**

The research activity and this interview will be conducted in accordance with the Market Research Society Code of Conduct, and the information you provide will be treated in accordance with data protection law. You have the right to withdraw your consent to our processing the information you provide or object to our processing of your information.

This interview is just to establish eligibility for the research project and will take around 5 minutes.

If you are eligible, you may be invited to attend an online interview which will last approximately 1 hour. You will be provided with a list of the topics to be covered in the interview ahead of time.

You will receive a £X incentive for participating in the interview. This payment will be paid via BACs by Roots Research if you qualify and attend the interview.

S1. Are you interested in proceeding further?

Yes	**CONTINUE TO S2**
No	THANK & CLOSE

S2. Before we start can I check whether you have taken part in any research or provided responses to any consultation for the Competition and Markets Authority in the past?

Yes	**CONTINUE TO S3**
No	**CONTINUE TO Q1**
Don't know/can't remember	**CONTINUE TO Q1**

S3. And was the research or consultation that you took part in related to the CMA study into the supply of smartphones and tablets, and associated software such as operating systems, app stores, browsers, and applications? This project was known as 'mobile ecosystems' and took place in 2021 and 2022.

Yes – I took part in the mobile ecosystems market study	THANK & CLOSE
Yes – I took part in the mobile ecosystems consultation	THANK & CLOSE
No – I didn't take part in either of these	**CONTINUE**

PLEASE ALSO REFER TO THE LIST OF PARTICIPANT NAMES PROVIDED AND DO NOT INCLUDE ANYONE ON THE LIST.

Q1a. Can I confirm that you currently work as a web developer?

Yes	1	**CONTINUE**
No	2	**THANK & CLOSE**

Q1b. And how long have you worked as a web developer?

Less than 2 years	1	**THANK & CLOSE**
2-5 years	2	**CONTINUE**
6-10 years	3	**CONTINUE**
More than 10 years	4	**CONTINUE**

AIM FOR A SPREAD OF CODES 2,3 AND 4.

Q1c. Thinking about your current role, which of the following best describes the types of website or web apps you are working on?

All of the websites and/or web apps I am working on are for use on mobile devices	1	**CONTINUE**
Most of the websites and/or web apps I am working on are for use on mobile devices		**CONTINUE**
Most/all of the websites and/or web apps I am working on are for use on other platforms, i.e., are not aimed at mobile devices		**THANK & CLOSE**

Q1d. And how much experience do you have of working on websites and/or web apps for use on mobile devices?

Less than 2 years	1	**THANK & CLOSE**
2-5 years	2	**CONTINUE**
6-10 years	3	**CONTINUE**
More than 10 years	4	**CONTINUE**

AIM FOR A SPREAD OF CODES 2,3 AND 4.

Q1e. Thinking about your current role, are all or most of the websites or web apps you are working on aimed at the UK market, i.e., are they for clients with a UK customer/user base?

Yes – all of the websites and/or web apps I am working on are aimed at the UK market	1	**CONTINUE**
Yes – most of the websites and/or web apps I am working on are aimed at the UK market		**CONTINUE**
No – all or most of the websites and/or web apps I am working on are aimed at the markets outside the UK		**THANK & CLOSE**

Q1f. Thinking about your work over the past 2 years, would you say you have mainly worked on websites, mainly on web apps or a fairly even mix of both?

Mainly websites	1	**CONTINUE**
Mainly web apps	2	**CONTINUE**
An even mix of both		**CONTINUE**

AIM FOR A MIX ACROSS THE DEPTHS AND CASE STUDIES.

Q1g. Thinking about your current role, which of the following best describes the type of web developer that you are?

I am a back-end developer	1	**THANK & CLOSE**
I am a front-end web developer	2	**CONTINUE**
I am a full-stack web developer		**CONTINUE**

ACROSS THE 40 DEPTHS WE NEED A MINIMUM OF 25 FULL-STACK AND 15 FRONT-END DEVELOPERS.

FOR THE CASE STUDIES ENSURE A MIX OF FULL-STACK AND FRONT-END DEVELOPERS.

Q1g. Are you personally based in the UK, i.e., do you live and work in the UK?

Yes, I am based in the UK	1	**CONTINUE**
No, I am based outside the UK	2	**CONTINUE**

AIM FOR 2-3 DEPTHS WITH WEB DEVELOPERS LIVING AND WORKING OUTSIDE THE UK.

Q2a. Which of the following best describes your current situation?

I work as part of an in-house web development team	1	**ASK Q2B**
I work for a web development agency	2	**ASK Q2B**
I work as a freelance web developer, getting contracts through an agency		**CONTINUE**
I am a freelance web developer working direct to my own clients		**CONTINUE**

FOR THE DEPTHS, MINIMUM OF 5 WITH THOSE WORKING FOR AN IN-HOUSE DEVELOPMENT TEAM AND MINIMUM OF 5 WORKING FOR A WEB DEVELOPMENT AGENCY.

FOR THE CASE STUDIES, ROUGHLY EVEN SPLIT BETWEEN IN-HOUSE WEB DEVELOPMENT TEAM AND WEB DEVELOPMENT AGENCY

ASK Q2B OF THOSE WORKING FOR IN-HOUSE WEB DEVELOPMENT TEAMS OR WEB DEVELOPMENT AGENCIES ONLY

Q2b. Which of the following best describes your current role?

I am a hands-on web developer without supervisory or managerial responsibilities	1	**CONTINUE**
I am a senior web developer working directly on web sites/apps but with responsibility for supervising the work of more junior web developers	2	**CONTINUE**
I have a senior role with responsibility for leading and managing a team of web developers	3	**CONTINUE**

FOR THE DEPTHS, NEED A MIX OF CODES 1 AND 2

FOR THE CASE STUDIES, WE NEED ONE PERSON WHO CODES 1 OR 2 PLUS ONE PERSON WHO CODES 3 FROM THE SAME ORGANISATION

Q3a. Thinking about the end clients for whom who you have developed websites/apps for mobile, which of the following sectors have you a) ever worked in and b) do you work in now?

	Ever worked in	**Work in now**
Marketing/Communications/Advertising/PR agencies	1	**1**
Social media	2	2
Internet Publishing and Broadcasting	3	3
Ecommerce and Online auctions	4	4
Gaming for mobile	5	5

WE ARE AIMING FOR A SPREAD OF SECTORS ACROSS THE RESEARCH.

Q3c. Thinking again about the end clients for whom who you have developed websites/apps for mobile, which of the following sizes of business have you ever developed websites/apps for?

Micro (1-9 employees)	1	**CONTINUE**
Small (10-49 employees)	2	**CONTINUE**
Medium (50-249 employees)	3	**CONTINUE**
Large (250+ employees)	4	**CONTINUE**

AIM FOR A SPREAD ACROSS THE DEPTHS AND CASE STUDIES

Q4. Which, if any, of these mobile ecosystems do you have a preference for working within?

I prefer to work within the Apple ecosystem (i.e. iOS operating system and Safari browser)	1	**CONTINUE**
I prefer to work within the Google ecosystem (i.e. Android and Chrome)	2	**CONTINUE**
I don't have a preference, I am happy working within either of them	3	**CONTINUE**

AIM FOR A SPREAD ACROSS THE DEPTHS AND CASE STUDIES. ENSURE WE HAVE A MINIMUM OF 5 DEPTHS WITH DEVELOPERS STATING A PREFERENCE FOR APPLE AND 5 WITH DEVELOPERS STATING A PREFERENCE FOR GOOGLE.

Q5. Finally, are you a member of any of the following organisations or communities?

ACT The App Association	1
Coalition for App Fairness	2
Developers Alliance	3
Marketers for an Open Web	4
Movement for an Open Web	5
Open Web Advocacy	6
Professional Publishers Association	7
Electronic Frontier Association	8
TIGA	9
No- I am not a member of any of the above	10

PLEASE MONITOR TO ENSURE THAT NO ONE ORGANISATION/COMMUNITY IS STRONGLY REPRESENTED IN THE RESEARCH. IF MORE THAN 1-2 FROM ANY ONE ORGANISATION/ COMMUNITY PLEASE ALERT JIGSAW AS MAY NEED TO EXCLUDE.

IF THE RESPONDENT QUALIFIES, EXPLAIN THE REQUIREMENTS FOR PARTICIPATING:

DEPTHS

S4. We would like to interview you online via Zoom. The interview will last up to an hour. We will send a list of the topics to be covered in advance of the interview. You will receive an incentive of £X as a thank you for your participation. Are you willing to take part?

Yes	1	**CONTINUE**
No	2	THANK & CLOSE

CASE STUDIES

S5. We would like to interview you online via Zoom. The interview will last up to an hour. We will send a list of the topics to be covered in advance of the interview. You will receive an incentive of £X as a thank you for your participation. We will also be conducting an interview with someone else within your organisation to enable us to have a fuller picture. Are you willing to take part?

Yes	1	**CONTINUE**
No	2	THANK & CLOSE

CONSENT: Thank you for agreeing to take part in our market research interview. In order to help protect your privacy, I will read out some terms and conditions that we will ask you to agree to and sign when you attend the interview.

S6. This study is being undertaken for market research purposes only by Jigsaw Research who adheres to the MRS code of conduct. The responses you give during the interview/discussion will be reported anonymously and in aggregate with responses from other research participants. The recordings may also be sent by secure transfer for transcription. Jigsaw Research will not disclose audio files/transcripts/interview notes gathered as part of a research exercise to CMA. Only anonymised quotes are used in our presentation and report to CMA. The information obtained about you during this project will be used solely for the purpose of this project.

Are you happy to proceed on this basis?

Yes	1	**CONTINUE**
No	2	THANK & CLOSE

S7. As part of our research process, and only with your express consent on the day, a member of the CMA team may wish to observe our interview with you, but we would take efforts to protect your confidentiality in this instance.

Are you happy to proceed on this basis?

Yes	1	**CONTINUE**
No	2	**CONTINUE**

DO NOT EXCLUDE IF UNWILLING TO BE OBSERVED – RECRUIT IF ELIGIBLE AND FLAG AS NO OBSERVATION OF INTERVIEW.

S8. Personal data is data that allows a living individual to be identified, either directly or indirectly. As part of this research, Jigsaw would like to make a digital recording of our interview with you for analysis purposes. We will securely delete the recording from our systems no later than 30 months after the interview.

Are you happy to proceed on this basis?

Yes	1	**CONTINUE**
No	2	THANK & CLOSE

S9. Please note that by giving your consent to participate in an interview, you consent to Roots Research Ltd and Jigsaw Research Ltd using and storing (processing) the personal data we have collected from you during your recruitment to the research and any further personal data you share with us during your interview. Roots Research Ltd and Jigsaw Research Ltd will use and store (process) your personal data for the purposes of this research project only.

Your personal data will not be shared with our client, the Competition and Markets Authority, in a way that would allow you to be individually identified. Please note that you are also consenting to the CMA processing aggregated data from which you cannot be individually identified, in the form of our report for the CMA on the findings from the research.

None of Roots Research Ltd, Jigsaw Research Ltd or the CMA will share your personal data with any third party, except in the very unlikely event that we are required to do so by law.

You have the right to access and rectify any data held on you by Roots Research or Jigsaw Research and to withdraw consent at any time or to object/restrict any processing of your data

We will securely delete all your personal data from our systems no later than 30 months after the interview.

Should you wish this information to be deleted earlier than this, or if you have any questions about how your personal information will be used, you can contact Jigsaw Research Ltd on 020 7291 0810 or by emailing datasecurity@jigsaw-research.co.uk.

If you have any concerns about Jigsaw's information rights practices, you can report it to the ICO via their website by calling 0303 123 1113.

Are you happy to proceed on this basis?

Yes	1	**CONTINUE**
No	2	THANK & CLOSE

S10. If we needed to contact you within 6 months of this study for any follow-up questions relating to this study specifically, is this OK?

Yes	1
No	2

RESPONDENT DETAILS

Name	
Address	
Phone	
Email	
Date of depth attending	

Reconfirm Day/Date/Time of the depth with respondent

INTERVIEWER DETAILS

Name	

INTERVIEWER DECLARATION

I declare that this interview has been conducted strictly in accordance with your specifications within the MRS Code of Conduct and with a person totally unknown to me.

Signed **(INTERVIEWER):** _____ Date: _____

4.2. Pre-interview task

Research with Web Developers

Thank you for agreeing to take part in our research with web developers; we appreciate you being willing to spend time talking with us.

As explained, we want to understand your experiences when developing websites and/or web apps for mobile devices (i.e., smartphones and tablets), and in particular your experiences of working with different browsers and browser engines which can be used to access those.

During the interview we will be asking you to describe in detail your day-to-day, hands-on experiences of developing websites and/or web apps for mobile devices. We will be asking you to feedback on which browser and browser engine features, functionalities and APIs:

A. Work well in helping you to develop websites and/or web apps that meet the requirements of your clients

B. Do not work well in helping you to develop websites and/or web apps that meet the requirements of your clients

It would be really helpful if, between now and the day of your interview, you had a think about the following key questions while you are going about your day-to-day development work:

Q1. What browser and browser engine features, functionalities or APIs work well; make your job easier, make it easier to optimise the website/web app for mobile devices?

Q2. What browser and browser engine features, functionalities or APIs work less well; make your job harder, make it harder to optimise the website/web app for mobile devices?

Q3. What browser and browser engine features, functionalities or APIs do you see as essential when developing websites and/or web apps for mobile devices, that you wouldn't like to live without?

Q4. Does the absence or presence of particular features in any one browser engine change how you develop for mobile devices in your work?

We are interested both in the 'big issues' and also in the smaller features you value or the niggles you experience when working with browsers and browser engines to develop for mobile devices.

If you were able to take some notes on the above for us to talk through during the interview, that would be really great.

Thanks in advance and look forward to talking with you.

4.3. Discussion guide – main fieldwork

Mobile Browsers MIR

Question Areas for Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines

Draft topic guide

Interview length – 1 hour

Introduction and warm up (c 10 mins)

Confirm recording for transcription purposes and review consent and confidentiality protocol

1. Can you tell me about your work as a web developer: how long you have worked as a developer, your main area of work, key client sectors and size…

   Ensure the following are covered as part of the discussion:

   - Check: developing for mobile is currently the main focus of their work and how long have worked in this area (minimum two year)
   - Check: type of developer specialism: Front-end/Back-end/Full-Stack or how they prefer to describe their specialism.
   - Check: developing websites and apps with focus on mobile devices.
   - Check: have developed websites and/or native apps and worked closely with the existing browser engine requirement of Apple's iOS and Android OS.
   - Check: awareness of current browser engine requirements when developing browsers on both iOS and Google. Are they aware that all mobile browsers on iOS – including Chrome – must use Safari's browser engine WebKit? Are they aware that mobile browsers on Android can use other browser engines – e.g. Chrome runs on Google's browser engine Blink on Android?
   - Check main client sectors: govt, broadcasting, retail, ecommerce, digital, social media and marketing agencies, internet publishing and broadcasting, tech sector, banking.
   - Check main size of client businesses: micro (1-9 staff), small (10-49), medium (50-249) and large businesses (250+).
   - Check if member of any developer associations or open web organisations. If so which and why joined:
     - ACT The App Association
     - Developers Alliance
     - Open Web Advocacy
     - Movt for an Open Web

2. How would you describe the focus of your work to an interested non-specialist?
    - Probe how much work is on developing and optimising websites and apps for mobile.
    - Probe on how long have worked on developing and optimising websites and apps for mobile.
    - Probe their personal focus in terms of browsers and browser engines, if they have one. Meaning whether they think of a specific browser and/or browser engine when developing and optimising websites and apps for mobile.
        - How did they end up with that focus? Was it deliberate? Was it what they learned first? Was it a business decision?
        - If they do not have one (i.e. they do not prioritise for any browser and/or browser engine in particular) probe rationale.
    - Probe services offered: web development, search engine optimisation, interface design, mobile web design, web app development, native app development and ecommerce.

Discussion of working on websites and apps for mobile (c 20 mins)

---

3. What do you see as the key trends in the last 5 or so years in your work on developing and optimising websites and web apps for mobile devices?
4. What do you see as the main challenges facing you in your work on developing and optimising websites and web apps for mobile devices? What is the top challenge? (if anything relates to browser choice/browser engine choice/Apple or Google don't probe at this stage – note and return to in later questions as relevant)
5. If you were speaking to someone who was interested in working in this area what would you say is good about developing websites and web apps for mobile?
6. What would you say was not so good or 'to be aware of' when working on developing websites/web apps for mobile?
    - What do you see as the advantages and disadvantages of developing a web app compared to a native app? Which is the biggest advantage/disadvantage of each?
    - Probe does this vary between types of app, e.g. more technically demanding vs less
    - Probe any other key differences between developing web apps v native apps.
    - Probe relative importance of websites, native apps and web apps based on own development work. If and how this has changed over time. If it has/is changing is this for specific types of applications e.g. gaming or a more general trend.
7. Do you tend to optimise your website development to ensure compatibility with specific browsers or browser engines on certain mobile devices? If so, how do you decide which browser or browser engines to optimise for? Which browsers do you most often prioritise to ensure compatibility and why?
    - Probe work with Apple's browser Safari on iOS

- Probe work with Google's browser Chrome on iOS and/or Android
- Probe work with Mozilla's browser Firefox on iOS and/or Android
- Probe work with other non-Google browsers on Android (e.g. Samsung Internet)
- Probe any change in prioritisation over time

8. Please explain briefly how you optimise a web app or website to ensure compatibility with a given browser (or browser engine).
    - Probe removing unsupported features.
    - Probe design considerations.
    - Probe testing and fixing bugs.
    - Probe designing/building in accordance with web standards.
    - Probe for resources used to help with cross-browser compatibility.
    - Probe for use of 'shims' or 'polyfills' to address compatibility issues.
9. What do you see as the key features distinguishing browsers on iOS vs Android?
    - Probe differences in functionality and/or standards supported.
    - Probe effort required for website or web app optimisation.
10. What do you see as the key features distinguishing the browsers you have most experience with?
    - Probe differences in functionality and/or standards supported.
    - Probe effort required for website or web app optimisation.
11. What are the key browser and browser engine features, functionalities or APIs that your website/web app would require to be rendered at its best?
12. Are there any features not currently supported by major browsers that you would like to see supported?
    - Probe: can you give examples to help explain how it would help you in your development work to have these features and how it would make your websites/web apps better.
    - Probe: any examples of key features currently supported by one browser but not another?
    - Probe: if this leads to using tactics to prompt users to switch browser for a better experience of website?
13. Are there any features currently supported by major browsers that you have decided not to use?
    - Probe: reason for decision e.g. costs, time, reach, browser's lack of scale. Examples.

Discussion of Android Ecosystem (c 10 mins)

Interviewer Note: rotate order so that broadly half interviewers discuss Android Ecosystem first and half Apple Ecosystem first. NB – some developers may not be able to discuss both in the same level of detail

---

14. In your experience, when you are developing websites and web apps with an Android environment in mind what are the advantages and disadvantages for you as a *developer*? Which is the biggest advantage/disadvantage?

    - Check ref is to Google's set-up with Chrome browser on Blink engine.
    - Check for examples

Themes which may emerge include: Familiarity/ease of use/speed/efficiency/cost/flexibility to be creative*/suits their technical skills and capabilities/helps with career prospects/Following browser standards*

15. What do you see as the advantages and disadvantages associated with developing websites or web apps targeted to an Android environment for your *clients*? Which is the biggest advantage/disadvantage?

Interviewer Note: in-house developers may use the term in-house clients or colleagues.

16. What do you see as the advantages and disadvantages for *consumers*? Which is the biggest advantage/disadvantage?

Themes which may emerge include: Familiarity/ease of use/Choice of browser engine/Following browser standards/New browser features/Speed/Capabilities/Privacy/Security/Web apps/Web compatibility/CSS compatibility/Progressive Web App features/Push Notifications/Full screen API/Bug Fixes/Update Cycle/Differentiation/Geolocations

    - Probe for examples

17. In your experience, are there any major differences – advantages or disadvantages – when you are developing for Android but working with Firefox running on the Gecko browser engine?

    - Probe for you as a developer ? For your clients ? For wider consumers?

18. How is your business affected by the current rules on which browser engines can be used on Android? Including which features are supported and standards implemented?

    - Probe are there any features, functionalities or APIs that your websites/web apps require to be rendered at their best which are currently missing support from Blink?]
    - Probe for each of: Security, Privacy, and Accessibility
    - Prob are there any issues which are specific to Blink (e.g. including bugs, missing features, functionalities or adherence to web standards) that require you to invest time or money in solving/addressing?

Themes which may emerge include: Cloud gaming/Web Apps/Choice/App stores/Security/Performance of browsers/Browser engine restrictions/API self-preferencing (meaning certain features and functionalities reserved to Google's own browser)/Compatibility and web standards compliance/Availability of Features on Blink/Speed on older phones/Support of web standards/publishing to web stores/DB implementations/Documentation/Update Logs

19. Thinking about your sector, are there wider effects from the rules on which browser engines can be used on Android in relation to features implemented and standards supported ?

    - probe for efficiencies and advantages.
    - probe for potential barriers and disadvantages.

20. Are there any positives associated with developing a web app as opposed to a native app on Android? Are there any drawbacks?

    - Probe development cost.
    - Probe speed of development
    - Probe ease of development
    - Probe functionality.
    - Probe performance.
    - Examples

Discussion of Apple Ecosystem (c 10 mins)

Interviewer Note: rotate order so that approx. half discuss Android Ecosystem first and approx. half Apple Ecosystem first. NB – some developers may not be able to discuss both in the same level of detail

---

21. In your experience, when you are developing websites or web apps with an iOS environment in mind what are the advantages and disadvantages for you as a developer? Which is the biggest advantage/disadvantage?

    - Check: ref is Safari on WebKit or Chrome on WebKit

Themes which may emerge include: Familiarity/ease of use/speed/efficiency/cost/flexibility to be creative/suits their technical skills and capabilities/helps with career prospects/Following browser standards

22. What do you see as the advantages and disadvantages associated with developing websites or web apps targeted to an iOS environment for your *clients*? Which is the biggest advantage/disadvantage?

Interviewer Note: in-house developers may use the term in-house clients or colleagues.

23. What do you see as the advantages and disadvantages for *consumers*? Which is the biggest advantage/disadvantage?

Themes which may emerge include: Familiarity/ease of use/Lack of browser engine choice/Following browser standards/New browser features/Speed/Capabilities/Privacy/Security/Restrictions on Web apps/Web compatibility/CSS compatibility/Progressive Web App features/Push Notifications/Full screen API/Bug Fixes/Update Cycle/Differentiation/Geolocations

    - Probe for examples

24. How is your business affected by the current rules on which browser engines can be used on iOS? Including which features are supported and standards implemented ?

    - Probe: are there any features, functionalities or APIs that your websites/web apps would require to be rendered at their best which are currently missing support from WebKit?
    - Probe for each of: Security, Privacy, and Accessibility
    - Probe are there any issues which are specific to WebKit (e.g. including bugs, missing features, functionalities or standards) that require you to invest time or money in solving/addressing?

Themes which may emerge include: Speed/Cost/Innovation/New types of services/Cloud gaming/Web Apps/Choice/App stores/Security/Performance of browsers/Browser engine ban/API self-preferencing (meaning certain features and functionalities reserved to Apple's own browser)/Compatibility and web standards compliance/Availability of Features on WebKit/Speed on older phones/Support of web standards/publishing to apple web store/DB implementations/Documentation/Update Logs

    - Probe have you noticed these issues changing – improving or otherwise – over time.

25. Thinking about your sector, are there wider effects from the rules on which browser engines can be used on iOS in relation to features implemented and standards supported?

- Probe for efficiencies and advantages.
- Probe for potential barriers and disadvantages.

26. Are there any positives associated with developing a web app as opposed to a native app on iOS Are there any drawbacks?
    - Probe development cost.
    - Probe speed of development
    - Probe ease of development
    - Probe functionality.
    - Probe performance.
    - Examples

Wrap up (c 10 mins)

27. Thinking about the experiences you have shared, do you have an accurate idea of how much it has cost you to ensuring web compatibility with both iOS and Android browsers?
    - Probe main factors contributing to this?
28. To what extent are the browser engines and browsers you prioritise developing for likely to change in the next 2 years?
    - Probe rationale (including if they do not prioritise any)
    - Probe any emerging market changes/technology changes e.g. emerging new browsers/use of AI
    - Probe any emerging impacts on their development work or planning
29. Reflecting on your experience would more choice within each ecosystem – so that browsers could pick between the 3 main existing browser engines – be beneficial or disadvantageous for you as a websites/web apps developer?
    - Probe rationale
    - Probe for examples of implications in their development work
30. Are there any positives with the situation as it is at the moment? (i.e. – with limited choice of browser engines on iOS, etc)
    - Probe on: ease, don't have to learn different systems, optimise for different browser engines, ensure compatibility with different browser engines, etc
    - Do you foresee any challenges or risks for you if within each ecosystem, browsers could choose more freely between the 3 main existing browser engines? Probe for perceived concerns, barriers to overcome or risks compared to the status quo and current level of choice.
    - Prompt for any final thoughts/feelings/opinions

Thank you/close with reminder on confidentiality protocol.

4.4. Discussion guide – follow-up fieldwork

Changes compared to the main discussion guide added from the interim analysis session just before fieldwork was paused, and then for the follow-up interviews are highlighted in grey

Mobile Browsers MIR

Question Areas for Qualitative Research with Developers on Mobile Browsers and Mobile Browser Engines

Draft topic guide – recontacted sample

Interview length – 45mins – 1 hour

Introduction and warm up (c 10 mins)

Introduction and reminder

Repeat broad introduction as per original guide.

- **Only read out if needed if asked about the case pause or why recontact phase is necessary:** If needed read out CMA paragraph about why the research was paused and restarted:
  - The CMA suspended its market investigation following a Competition Appeal Tribunal (CAT) order that suspended the CMA's market investigation with effect from 31 March 2023. This order followed a legal challenge by Apple on the CMA's decision to launch the market investigation. On 30 November 2023, the Court of Appeal determined that the CMA's decision to launch the investigation was lawful and set aside the CAT's judgment. Consequentially, the market investigation recommenced on **24 January 2024**. The statutory deadline for the Market Investigation is March 2025
- Explain that we are interested in reviewing what they said last time, overall to see if and how the situation and their views have changed since the research was paused, as well as asking some specific new questions
- Our approach will generally be to ask you (for each topic we want to cover) what has changed (if anything) then to refer back to your old answers to get into the detail and help jog your memory

Confirm recording for transcription purposes and review consent and confidentiality protocol

- Has anything changed about your work situation since we last spoke? What?
  - Prompt with previous answers

1. Can you tell me about your work as a web developer: how long you have worked as a developer, your main area of work, key client sectors and size

   Ensure the following are covered as part of the discussion:

   - Check: developing for mobile is currently the main focus of their work and how long have worked in this area (minimum two year)
   - Check: type of developer specialism: Front-end/Back-end/Full-Stack or how they prefer to describe their specialism.
   - Check: developing websites and web apps with focus on mobile devices.
   - Check: have developed websites and/or native apps and worked closely with the existing browser engine requirement of Apple's iOS and Android OS.
   - Check: awareness of current browser engine requirements when developing browsers on both iOS and Google. Are they aware that all mobile browsers on iOS – including Chrome – must use Safari's browser engine WebKit? Are they aware that mobile browsers on Android can use other browser engines – e.g. Chrome runs on Google's browser engine Blink on Android? (NB. Blink may also be referred to as Chromium or V8)
   - Check main client sectors: govt, broadcasting, retail, ecommerce, digital, social media and marketing agencies, internet publishing and broadcasting, tech sector, banking.
   - Check main size of client businesses: micro (1-9 staff), small (10-49), medium (50-249) and large businesses (250+).
   - Check if member of any developer associations or open web organisations. If so which and why joined:
     - ACT The App Association
     - Developers Alliance
     - Open Web Advocacy
     - Movt for an Open Web

   Has the focus of your work changed or shifted since we last spoke? What has happened?

   - Prompt with previous answers as needed
   - This is what you broadly said last time – has any of this changed? How?

2. How would you describe the focus of your work to an interested non-specialist?

   - Probe how much work is on developing and optimising websites and web apps for mobile.
   - Probe on how long have worked on developing and optimising websites and web apps for mobile.
   - Probe their personal focus in terms of browsers and browser engines, if they have one. Meaning whether they think of a specific browser and/or browser engine when developing and optimising websites and web apps for mobile.

- How did they end up with that focus? Was it deliberate? Was it what they learned first? Was it a business decision?
- If they do not have one (i.e. they do not prioritise for any browser and/or browser engine in particular) probe rationale.
- If they mention a browser in general terms here or elsewhere e.g. ‘developing for Chrome’, probe on the device/ecosystem they are talking about e.g. IOS/Android etc

- Probe services offered: web development, search engine optimisation, interface design, mobile web design, web app development, native app development and ecommerce.

Discussion of working on websites and web apps for mobile (c 20 mins)

Have you seen any new trends emerging in your work on developing and optimising websites since we last spoke?

3. What do you see as the key trends in the last 5 or so years in your work on developing and optimising websites and web apps for mobile devices?

   Are you seeing any new challenges emerge?

   - Are any of the old challenges still there? (prompt with previous answers)

4. What do you see as the main challenges facing you in your work on developing and optimising websites and web apps for mobile devices? What is the top challenge? (if anything relates to browser choice/browser engine choice/Apple or Google don't probe at this stage – note and return to in later questions as relevant)

   If AI mentioned, probe how it's being used. E.g., how do they use ChatGPT (or similar)?

   If you were speaking to someone who was interested in working in this area what would you say is good about developing websites and web apps for mobile?

5. What would you say was not so good or 'to be aware of' when working on developing websites/web apps for mobile?

   Has anything changed in regards to the advantages vs disadvantages of developing a web app vs a native app? (prompt with previous answers)

6. What do you see as the advantages and disadvantages of developing a web app compared to a native app? Which is the biggest advantage/disadvantage of each?

   - Probe does this vary between types of app, e.g. more technically demanding vs less
   - Probe any other key differences between developing web apps v native apps.
   - *Is shift away from native apps due to better developer tools or because web apps have become as good as native apps?*
   - Probe relative importance of: websites, native apps and web apps based on own development work. If and how this has changed over time. If it has/is changing is this for specific types of applications e.g. gaming or a more general trend.

   Has anything changed in regards how you work with different browsers?

   - Probe around optimising website development
   - Browsers they are optimising for
   - Prompt with previous answers

7. Do you tend to optimise your website development to ensure compatibility with specific browsers or browser engines on certain mobile devices? If so, how do you decide which browser or browser engines to optimise for? Which browsers do you most often prioritise to ensure compatibility and why?

- Probe work with Apple's browser Safari on iOS
- Probe work with Google's browser Chrome on iOS and/or Android
- Probe work with Mozilla's browser Firefox on iOS and/or Android
- Probe work with other non-Google browsers on Android (e.g. Samsung Internet)
- If they mention a browser in general terms here or elsewhere e.g. 'developing for Chrome', probe on the device/ecosystem they are talking about e.g. IOS/Android etc
- Probe any change in prioritisation over time:
  - If mention issues with browsers in the past or with browsers they do not work with any more (e.g. Internet Explorer), probe which browsers and nature of the problem/challenges

Has how you do this changed at all?

- Probe around standards being used?
- Prompt with previous answers

8. Please explain briefly how you optimise a web app or website to ensure compatibility with a given browser (or browser engine):
   - Probe removing unsupported features.
   - Probe design considerations.
   - Probe testing and fixing bugs.
   - Probe designing/building in accordance with web standards – probe for specific web standards being referred to and why they are important to them/their websites
   - Do you make use of web standards when building websites/web apps?
     - Probe for: awareness of W3C standards for coding best practice (HTML; CSS; XML)
     - WCAG accessibility standards
   - IF YES: do you find these helpful? Why/why not?
     - How are these applied in your work?
     - How do you check that your code is standards compliant?
   - Do you think there are areas of website/web app development where you think standards are missing, or could be more detailed?
   - Probe for resources used to help with cross-browser compatibility.
   - Probe for use of 'shims' or 'polyfills' to address compatibility issues.
   - *Probe/listen out for any mentions of in-app browsers – what impact does this have on the code they have to write? Differences between IOS vs Android here?*

- Has your view of iOS or Android browsers changed at all?

- Have your perceptions of the differences between the two changed?
- What about the key differences between the browsers you use most often?
- Have web standards changed?
  - Prompt with previous answers

9. What do you see as the key features distinguishing browsers on iOS vs Android?
   - Probe differences in functionality and/or standards supported – probe for specific web standards being referred to and why they are important to them / their websites
   - Probe effort required for website or web app optimisation
   - *If IOS mentioned as more protective/more stipulations when developing web apps for IOS – Probe on what this means?*
   - *Do differences between IOS and Android/Google extend from mobile to desktop?*
   - *If 'part of the job' mentioned – probe on what this means. Good/bad and why?*
10. What do you see as the key features distinguishing the browsers you have most experience with?
    - Probe differences in functionality and/or standards supported – probe for specific web standards being referred to and why they are important to them/their websites
    - Probe effort required for website or web app optimisation.
11. What are the key browser and browser engine features, functionalities or APIs that your website/web app would require to be rendered at its best?
    - Has your view on whether there are any features missing/not supported by the major browsers changed?
      - Prompt with previous answers

12. Are there any features not currently supported by major mobile browsers that you would like to see supported?
    - Probe: can you give examples to help explain how it would help you in your development work to have these features and how it would make your websites/web apps better.
    - Probe: any examples of key features currently supported by one browser but not another?
    - Probe: if this leads to using tactics to prompt users to switch browser for a better experience of website?
13. Are there any features currently supported by major browsers that you have decided not to use?
    - Probe: reason for decision e.g. costs, time, reach, browser's lack of scale. Examples.

Discussion of Android Ecosystem (c 10 mins)

Interviewer Note: rotate order so that broadly half interviewers discuss Android Ecosystem first and half Apple Ecosystem first. NB – some developers may not be able to discuss both in the same level of detail

---

Overall, has your view of the Android ecosystem changed at all since we last interviewed you?

- For each numbered question, ask:
- Has your view of [question] changed?
- Prompt with answers from last time

14. In your experience, when you are developing websites and web apps with an Android environment in mind what are the advantages and disadvantages for you as a *developer*? Which is the biggest advantage/disadvantage?

    - Check ref is to Google's set-up with Chrome browser on Blink engine. (NB. Blink may also be referred to as Chromium or V8)
    - Check for examples
    - *Probe on what is meant by web developer tools.*

    Themes which may emerge include: Familiarity/ease of use/speed/efficiency/cost/flexibility to be creative/suits their technical skills and capabilities/helps with career prospects/Following browser standards – probe for specific web standards being referred to and why they are important to them/their websites

    E.g. responsiveness, videos, animation, sliders, hovering etc…. probe for as many examples as possible

15. What do you see as the advantages and disadvantages associated with developing websites or web apps targeted to an Android environment for your *clients*? Which is the biggest advantage/disadvantage?

    Interviewer Note: in-house developers may use the term in-house clients or colleagues.

16. What do you see as the advantages and disadvantages for consumers? Which is the biggest advantage/disadvantage?

    Themes which may emerge include: Familiarity/ease of use/Choice of browser engine/Following browser standards (probe for specific web standards being referred to and why they are important to them/their websites)/New browser features/Speed/Capabilities/Privacy/Security/Web apps/Web compatibility/CSS compatibility/Progressive Web App features/Push Notifications/Full screen API/Bug Fixes/Update Cycle/Differentiation/Geolocations

    - Probe for examples

17. In your experience, are there any major differences – advantages or disadvantages – when you are developing for Android but working with Firefox running on the Gecko browser engine?

- Probe for you as a developer. For your clients ? For wider consumers?

18. Are you aware of any restrictions for using certain browser engines on Android? Including which features are supported and standards implemented?

    - Probe are there any features, functionalities or APIs that your websites/web apps require to be rendered at their best which are currently missing support from Blink?]
    - Probe for each of: Security, Privacy, and Accessibility
    - Prob are there any issues which are specific to Blink (e.g. including bugs, missing features, functionalities or adherence to web standards) that require you to invest time or money in solving/addressing? – probe for specific web standards being referred to and why they are important to them/their websites

    Themes which may emerge include: Cloud gaming/Web Apps/Choice/App stores/Security/Performance of browsers/Browser engine restrictions/API self-preferencing (meaning certain features and functionalities reserved to Google's own browser)/Compatibility and web standards compliance – probe for specific web standards being referred to and why they are important to them/their websites/Availability of Features on Blink/Speed on older phones/Support of web standards/publishing to web stores/DB implementations/Documentation/Update Logs

19. Thinking about your sector, are there wider effects from the rules on which browser engines can be used on Android in relation to features implemented and standards supported?

    - Probe for efficiencies and advantages.
    - Probe for potential barriers and disadvantages.
    - Probe for specific web standards being referred to and why they are important to them/their websites

20. Are there any positives associated with developing a web app as opposed to a native app on Android? Are there any drawbacks?

    - Probe development cost.
    - Probe speed of development
    - Probe ease of development
    - Probe functionality.
    - Probe performance.
    - Examples

Discussion of Apple Ecosystem (c 10 mins)

Interviewer Note: rotate order so that approx. half discuss Android Ecosystem first and approx. half Apple Ecosystem first. NB – some developers may not be able to discuss both in the same level of detail

---

Overall, has your view of the Apple ecosystem changed at all since we last interviewed you?

- For each numbered question, ask:
- Has your view of [question] changed?
- Prompt with answers from last time

21. In your experience, when you are developing websites or web apps with an iOS environment in mind what are the advantages and disadvantages for you as a *developer*? Which is the biggest advantage/disadvantage?

    - Check: ref is Safari on WebKit or Chrome on WebKit
    - *Probe on what is meant by web developer tools.*

    Themes which may emerge include: Familiarity/ease of use/speed/efficiency/cost/flexibility to be creative/suits their technical skills and capabilities/helps with career prospects/Following browser standards – probe for specific web standards being referred to and why they are important to them/their websites

    E.g. responsiveness, videos, animation, sliders, hovering etc…. probe for as many examples as possible

22. What do you see as the advantages and disadvantages associated with developing websites or web apps targeted to an iOS environment for your *clients*? Which is the biggest advantage/disadvantage?

    Interviewer Note: in-house developers may use the term in-house clients or colleagues.

23. What do you see as the advantages and disadvantages for *consumers*? Which is the biggest advantage/disadvantage?

    Themes which may emerge include: Familiarity/ease of use/Lack of browser engine choice/Following browser standards – probe for specific web standards being referred to and why they are important to them/their websites/New browser features/Speed/Capabilities/Privacy/Security/Restrictions on Web apps/Web compatibility/CSS compatibility/Progressive Web App features/Push Notifications/Full screen API/Bug Fixes/Update Cycle/Differentiation/Geolocations

    - Probe for examples

24. How is your business affected by the current rules on which browser engines can be used on iOS? Including which features are supported and standards implemented ?

    - Probe: are there any features, functionalities or APIs that your websites/web apps would require to be rendered at their best which are currently missing support from WebKit?

- Probe for each of: Security, Privacy, and Accessibility
- Probe are there any issues which are specific to WebKit (e.g. including bugs, missing features, functionalities or standards) that require you to invest time or money in solving/addressing? – probe for specific web standards being referred to and why they are important to them/ their websites

Themes which may emerge include: Speed/Cost/Innovation/New types of services/Cloud gaming/Web Apps/Choice/App stores/Security/Performance of browsers/Browser engine ban/ API self-preferencing (meaning certain features and functionalities reserved to Apple's own browser)/Compatibility and web standards compliance – probe for specific web standards being referred to and why they are important to them/their websites/Availability of Features on WebKit/Speed on older phones/Support of web standards/publishing to apple web store/ DB implementations/Documentation/Update Logs

- Probe have you noticed these issues changing – improving or otherwise – over time.

25. Thinking about your sector, are there wider effects from the rules on which browser engines can be used on iOS in relation to features implemented and standards supported?
    - Probe for efficiencies and advantages.
    - Probe for potential barriers and disadvantages?
    - Probe for specific web standards being referred to and why they are important to them/their websites
26. Are there any positives associated with developing a web app as opposed to a native app on iOS Are there any drawbacks?
    - Probe development cost.
    - Probe speed of development
    - Probe ease of development
    - Probe functionality.
    - Probe performance.
    - Examples

Wrap up (c 10 mins)

Overall, how much do you think it has cost you to ensure compatibility with both iOS and Android browsers?

- Probe: cost in terms of time, money or any other cost to you personally
- Has this increased since we last spoke? Why?
- Prompt with answers from last time

27. Thinking about the experiences you have shared, do you have an accurate idea of how much it has cost you to ensuring web compatibility with both iOS and Android browsers?
    - Probe main factors contributing to this? Listen out for differences by co.size
    - *Ongoing costs or set up costs?*

    As far as you are aware, to what extent do you think the browser engines and browsers you develop for likely to change in the next 2 years?

    - How will AI affect this?
    - Prompt with answers from last time

28. To what extent are the browser engines and browsers you prioritise developing for likely to change in the next two years?
    - Probe rationale (including if they do not prioritise any)
    - Probe any emerging market changes/technology changes e.g. emerging new browsers/use of AI
    - Probe any emerging impacts on their development work or planning

    Would more choice within each ecosystem be beneficial or advantageous to you? Why?

    - Has this changed since we last spoke? How Prompt with answers from last time

29. Reflecting on your experience would more choice within each ecosystem – so that browsers could pick between the 3 main existing browser engines – be beneficial or disadvantageous for you as a websites/web apps developer?
    - *Probe rationale*
    - *Probe for examples of implications in their development work*

    Do you see any positives with the limited choice that you have at the moment? Why?

    - Has this changed since we last spoke? How? Prompt with answers from last time

30. Are there any positives with the situation as it is at the moment? (i.e. – with limited choice of browser engines on iOS, etc)

- Probe on: ease, don't have to learn different systems, optimise for different browser engines, ensure compatibility with different browser engines, etc
- Do you foresee any challenges or risks for you if within each ecosystem, browsers could choose more freely between the 3 main existing browser engines? Probe for perceived concerns, barriers to overcome or risks compared to the status quo and current level of choice.
- *If short term pain, long term gain sentiment expressed – probe on how they imagine this happening?*
- *What does 'good' look like?*
- *Prompt for any final thoughts/feelings/opinions*

Overall, having now revisited what we discussed last time…

- *What has been the biggest change, if any? How has this affected you?*
- *What's better?*
- *What's worse?*
- *Do you feel you have more or less choice now?*

Thank you/close with reminder on confidentiality protocol.

MRS Awards 2022 Finalist Business-to-Business Research

MRS Awards 2020 Winner Innovation of the Year

MRS Awards 2019 Winner Global Agency of the Year

MRS Inclusion Pledge

THANK YOU

Jigsaw Research Ltd
3rd Floor, 53 Parker Street
London WC2B 5PT
Tel: +44 (0)20 7291 0810
Web: www.jigsaw-research.co.uk
Email: info@jigsaw-research.co.uk

Registered in England no.: 3650224
Registered Address: 11 Soho Street
London W1D 3AD
VAT no.: 326 011 940