**Understanding Design Decision Motivations in Software Development**

A Research Paper submitted to the Department of Engineering and Society

Presented to the Faculty of the School of Engineering and Applied Science
University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree
Bachelor of Science, School of Engineering

**Stuart Paine**
Spring 2023

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Advisor
MC Forelle, Department of Engineering and Society

## Introduction

As perhaps one of the most successful and highly regarded tech entrepreneurs of all time, Bill Gates claims that "Microsoft's failure with Windows Phone remains his biggest regret" (Jones, 2021). Although hindsight is 20/20, it doesn't take perfect vision to see that smartphones have become ubiquitous within modern developed societies and enjoy continual technological refinement and functional advancement. The Windows Phone is just one example of many failures by smartphone manufacturers and operating systems to compete well enough to survive over the last couple decades. This also plays into the polarizing rise of Android and iOS as the two mainstay mobile operating systems.

The current landscape of smartphone operating systems is quite binary with Android and iOS as the two most prominent by a large margin. Apple's iOS has the majority of the mobile phone operating system market share within the United States, whereas Android dominates globally (Mobile Operating System Market Share United States Of America, 2022). Although each operating system (Android & iOS) has areas where it may somewhat outperform the competitor, tests have not shown either operating system to be holistically more efficient than the other (Győrödi et. al., 2017). Other more obscure mobile operating systems, such as Windows and KaiOS hold miniscule market shares with under 1% combined both globally and domestically (Mobile Operating System Market Share United States Of America, 2022).

Given the ubiquity of mobile devices in modern developed societies, web traffic from mobile devices is close to overcoming traffic from desktop computers and is continuing to rise in popularity (Bouchrika, 2022). This idea of mobile-first computing, referring to one's tendency to use one or more mobile devices as their primary means of computing, has necessitated software design ideologies that cater to the typically smaller screens and differing functionalities of

mobile devices. A rise in this mobile-first design strategy has led to software that is more friendly and easier to use across various screen sizes.

With an impressive 85% of Americans now owning a smartphone of some kind, not to mention another 12% that own a more rudimentary cellphone, it is safe to say that the vast majority enjoys keeping a mobile computing device at arm's reach (Mobile Fact Sheet, 2021). This prominence of mobile device usage is compounded by the fact that "as of August 2022, 53.74 percent of the total web visits are currently mobile, compared to 46.26 percent coming from desktops" (What Percentage of Internet Traffic Is Mobile?, 2022). This difference in usage is also likely to keep growing, as "the market for mobile computing devices across the globe is anticipated to develop at the fastest rate over the next few years" (Data Bridge Market Research, 2022). It is imperative to design software with target user bases and usage circumstances in mind because deciding upon which operating system or systems to support essentially determines which groups of users will be able to effectively or conveniently utilize the software if at all. This is analogous to an author or publisher choosing the language or languages in which to release a book—different groups of readers will be affected differently given each choice.

Developer design decisions, including release platforms and beyond, are motivated by a variety of factors under unique circumstances in each instance that inevitably cause inequity in justly serving the needs of all end users and other stakeholders. Different use cases call for different platforms, with Android, iOS, and responsive web applications as the three main platforms each with their own set of features and characteristics. I will analyze instances of development within which specific causal factors, or a lack thereof, led to certain design decisions and thus resulted in identifiably respective consequences for better or worse. I began the research process by drawing on past experiences working on team-based software

development projects to identify general avenues of thought to further investigate and contextualize through the collection and study of secondary sources. The analysis will show that gaps exist in the process of catering software to all eventual end users, and that careful consideration of environmental conditions is needed on a case-by-case basis. Careful and thorough planning should be done before creating any consumer-oriented software product while thinking through the effects of all design choices. Although there is arguably no such thing as a perfect design, developers can certainly strive to do better.

**Literature Review**

There are two main ways through which most software applications are presented to the end user—web-based applications and device-native applications (Montecuollo, 2014). Web-based applications are usually able to suit a wider variety of users since all that is required to use them is a web browser, given that developers have taken care to account for the possibility of usage across different devices through responsive design (Berry, 2021). However, among other pros and cons of development for both sides, natively designed software applications often benefit from performance and functional advantages (Holzer & Ondrus, 2012).

With a multitude of software frameworks and deployment platforms available for developers to choose to cater to and release applications for, recent cross-platform development methodologies have made it easier to develop and release software for multiple platforms at once. Currently popular frameworks have made it much easier for developers to effectively design and deploy web applications that run platform-agnostically in a responsive manner across personal computers and mobile devices alike (Shahzad, 2017). Not only has it become easier to deploy web applications that can run platform agnostically, but cross-platform mobile application development using frameworks such as React Native and Flutter make it easier to

deploy device native mobile applications to both iOS and Android from a single codebase (Fentaw, 2020). Many earlier mobile cross-platform development frameworks such as Phonegap, released by Nitobi in early 2008, provided a development model for releasing apps that were technically native, yet all functionality was achieved through an embedded web browser within the app which defeats the whole purpose of providing a native experience (Hartmann et al., 2011). More recent, modern advancements like React Native have tremendously improved the experience of both developers and end users.

Within the elicitation and gathering of project requirements in the first stage of the software development process, one of the first topics that developers must decide upon is what platform they will be developing their product for ("6 Basic Steps of the Software Development Process (2022 Updated)," 2017). And inevitably, limitations in development team resources such as time or knowledge base, the directed interests of stakeholders, or other factors lead to software products being developed for certain platforms but not others (Dziuba, 2021). This means that many devices and thus many users are rendered unable to use the software. However, although common undertones of judgmental perceptions or stigmas exist regarding users of these mobile operating systems, users of Android and iOS may not be as different as people think. According to recent research, there is little difference in overall measurable aspects of personality between iOS and Android smartphone users (Götz et. al, 2017). On the contrary, research conducted in 2013 reported that "persons that said that the opinion of friends, family and colleagues affected them when choosing their smartphone" found themselves more "brand-aware" and "more likely to have an iPhone" to fit the aforementioned stereotype(s) (Benenson et al., 2013). Nonetheless, brand loyalty has been identified among owners of Apple and Samsung devices, thus resulting in de-facto loyalty to the iOS and Android mobile operating systems (Kim et. al., 2019).

Despite responsive web applications or cross-platform mobile applications appearing to be versatile solutions, different use cases necessitate different design schemas. Industrial progress in terms of technological advancements, referring to smartphones and other computing devices in this case, available for utilization by the modern consumer has had an extraordinary impact on society (Morison, 1980). However, there must be an emphasis on the user of said machinery to take advantage of the machine being used to best serve their own purposes. If a user doesn't have the ability to take advantage of said technology, then they cannot reap the rewards of that extraordinary impact (Morison, 1980). In other words, as powerful as software may be, it is still important to construct designs with specific use cases in mind. Each piece of software will have a unique usage landscape and different use cases that call for different software requirements and capabilities to be fulfilled, determining whether a mobile or web app would be most appropriate, for example (Turner-McGrievy et. al., 2016). There are many factors that must be considered when choosing web or mobile, such as the need for touchscreen interactivity, need for push notifications, etc. that help determine which system would be more useful on a case-by-case basis (Turner-McGrievy et. al., 2016). For example, if you need to be able to passively deliver push notifications to the user, this can be easily achieved with mobile applications but not directly through applications on the web. Choosing to develop for the wrong platform can seriously hinder performance and usability.

Latour's actor network theory (ANT) framework is useful in analyzing the factors behind how software succeeds or fails in properly catering to the needs of all users. ANT proposes that many elements of society, human and non-human, are connected in more ways than may be immediately apparent to the naïve investigator (Latour, 1996). Elements can be viewed as actors, networks, or both depending on the context of analysis. Actors and networks are interrelated, and

their relationships affect others, both directly and indirectly. Actors are defined as entities that act upon or influence others, not necessarily human, while a network describes a relationship between actors that is both defined by and helps define how actors interact with each other. The main human actors at the heart of this analysis include software developers, device manufacturers, and end users, while essential non-human actors consist of computing devices, software applications, release platforms, and development frameworks. This ANT framework can be used to deftly encapsulate the elements within and relationships between software developers, their development teams, and end users as well as other stakeholders in the design and development processes.

**Methods**

I have gathered secondary sources, primarily academic journal articles concerning cases where there are clear motivations outlined for how software development projects have been carried out to serve certain motivational agendas and why the conscious design choices in that process were valued and made. The scope of this research should not predate the mass introduction of smartphones to society, meaning I have focused on cases within roughly the last fifteen years with an emphasis on newer research since the technological landscape is continuously evolving. I have examined the motivations behind and the effects resulting from differences in software platform availability and other design and usage factors with a focus on the interactions between different players using actor network theory. Although it is rather straightforward to identify actors and networks in themselves, the truly insightful findings lie within the relationships between them.

**Analysis**

Design decisions are sometimes motivated by more extrinsic factors outside of the best wishes of the consumer. Monetary gain for parties responsible for developing and ultimately selling software and hardware is a great example of a motivating factor that causes inequity for obvious reasons well beyond the simple need for a company to stay afloat. Take the concept of digital or device ecosystems, for example: "Leading companies are increasingly offering an interconnected set of services" which encourage users to use software or hardware within a given family of products backed by a common entity (Dietz et al., 2020). Common device ecosystems include Amazon Echo devices, as well as Apple's conglomerate of iPhones, iPads, MacBooks, etc. These ecosystems lock interactivity within a confined usage space and leave users with less freedom in choosing their devices--they become trapped within that network. Once a consumer as an actor has purchased a device within a given ecosystem, they may tend to feel stuck within that network unless they wish to abandon ship and forfeit their initial purchase to replace it with an alternative that is able to interact with other devices outside of that network. This follows the idea of prescription under Latour's actor network theory, in that this pseudo-constraining of capitalistic freedom is "behavior *imposed* back onto the [consumer] by nonhuman delegates" in the form of ecosystems as networks (Latour, 1996). These ecosystems stem from conscious design decisions by manufacturers and innately restrict the freedom of users who wish to reasonably venture outside of the advertiser's dream of their logo singularly plastered throughout all facets of their users' lives.

In other cases, design decisions are sometimes necessitated by limitations regarding development or other infrastructural challenges. In terms of infrastructural challenges, problems often arise in how different software or hardware components are able to interface with each

other. Although it is often possible to solve interfacing challenges with enough work, it is often not worth the time and effort to try to mesh together different networks—perfectly fitting the idea of scripts that constrain relations between different parties under actor network theory. Different services offered by cloud service providers are a prime example of this idea. Amazon Web Service (AWS) services naturally work in tandem with one another, as do those of Google Cloud Platform (GCP). However, it becomes more difficult to interface services from one provider with those of another without extra work that could and probably should be avoided with a better infrastructural design of the different components (Google, n.d.). In this same scope lies the case of Apple's refusal to provide FaceTime or the coveted blue iMessage text bubbles to users outside of their own device ecosystem. Some may assume that these shortcomings stem out of unfortunate or physically unavoidable software interfacing limitations, but this is not the case. Although interfacing challenges may pose developmental hurdles, Apple chooses to create an in-crowd versus out-crowd dynamic for their own self-realized benefit and willingly refuses to allow FaceTime and iMessage's blue bubbles to be had by users of Android and other systems (Higgins, 2022). This is done to try to force the end users as actors to repetitively subscribe to their products while limiting users from subscribing to others. Given that the Android operating system is available on devices at price points ranging far below those of the cheapest Apple devices, consumers who care about maintaining what they may see as the status quo in owning an iPhone or iPad may feel left out if they're unable to make those larger purchases.

Design choices are not always clear cut; some choices have a much more apparent right and wrong or better and worse, while others do not. Let's say, for example, that the National Health Department of India wants to develop a mobile application for their citizens to keep track of their personal medical records. Ideally, great software would be available for both the Android

and iOS mobile operating systems. However, developer and other resource restrictions could realistically lead to the prioritization of one platform over the other with some users totally unable to access the software as a result. Given the government should have the best interest of its citizens as users in mind and healthcare is deemed vitally important, inclusivity and ease of use should be prioritized. Given the popularity of mobile computing, a mobile app or responsive web app should be used to cater to many groups of users (Bouchrika, 2022). This all goes to say that Android should absolutely be prioritized over iOS, if necessary, given that Android dominates mobile OS market share of India with almost 96% (Mobile Operating System Market Share India, n.d.). The prioritization, if necessary, of Android for this example is quite clear cut given that the usage tendencies and ownership figures have prescribed it upon the imaginary development team in this scenario through Latour's concept of prescription (Latour, 1996). However, such decisions are not always made so easily. Releasing a novel mobile game in the U.S. with closer to split market share may make it harder to decide between Android and iOS if forced to choose one (Mobile Operating System Market Share United States Of America, 2022).

Developers sometimes blatantly fail to understand or base design decisions on the needs of the end user or other relevant stakeholders, whether through a voluntary or involuntary ignorance of the prescription. There exist many bloated layers of human actors in the lengthy software development process network, between software developers, project managers, eventual end users, other stakeholders, and more. Gaps in understanding of what is wanted or needed inevitably form and are perpetuated by incorrect or lazy assumptions made by actors at any stage within the overall chain of responsibility within any development project. Although each level of actor in the development process may be connected to their adjacent actor or actors, that connection may only span one level in the hierarchy instead of providing a more connected

network with open feedback loops from start to finish. The degree of misunderstanding can easily become exacerbated by the snowball effect over the course of development, similar in fashion to the age-old children's game of telephone (Clinton, 2023). However, the popularization of agile development methodologies has helped combat these gaps in understanding and increased the flexibility of design and development (Alsaqqa et al., 2020). With an emphasis on requirements gathering and overall flexibility in design allowing for changes at any given point in the design process, agile methodologies have certainly taken a step in the right direction to try to solve these design inequalities.

**Conclusion**

By this point, it should be evident that design decisions are motivated by a myriad of factors, are quite impactful, and should be carefully considered on a case-by-case basis within any software development project. Developers may need to look more closely into their target users to better understand their needs and more carefully base design decisions on those needs or other perceived goals and intents. Future research could develop a logical framework for use in investigating the wants and needs or other characteristics of users to help convey to developers what design choices would best serve them. This would include potential ways to close the gap between developers' understanding of how design decisions can be made to best cater to their intended users. The rise in popularity of agile, responsive, and cross-platform development strategies paint a promising picture leading the way into the future of creating more equitable and better serving software.

Furthermore, this suggested emphasis on the importance of design factors spans well beyond software products into many other user-oriented design practices across the greater realm of engineering. Motivations are innately inescapable of bias. However, a great first step in

working toward more equitable software and other consumer products is to analyze the effects of paths taken by developers and designers within current popular culture to identify both shortcomings and successes in terms of serving users. Fortunately, it seems like the fundamental reliance modern developed societies have bound themselves to in computing technology has begun to increasingly motivate this step of self-reflection to aid future work in achieving a better state of understanding usability. Perhaps developers could start by placing a greater emphasis on considering how their viewpoints are vastly obscured insofar the insight of their development knowledge leads to biased assumptions motivating or reinforcing harmful design choices.

**References**

*6 basic steps of the software development process (2022 updated)*. (2017, February 23). Manifera. https://www.manifera.com/6-basic-steps-software-development-process/

Alsaqqa, S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. *International Journal of Interactive Mobile Technologies (IJIM), 14*(11), 246. https://doi.org/10.3991/ijim.v14i11.13269

Benenson, Z., Gassmann, F., & Reinfelder, L. (2013). Android and IOS users' differences concerning security and privacy. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. https://doi.org/10.1145/2468356.2468502

Berry, S. (2021, February 26). *What Is Responsive Web Design? (The Non-Developer's Cheat Sheet).* WebFX. https://www.webfx.com/blog/web-design/what-is-responsive-web-design/

Bouchrika, I. (2022, December 20). *Mobile vs desktop usage statistics for 2023.* Research.com. Retrieved March 1, 2023, from https://research.com/software/mobile-vs-desktop-usage

Clinton, B. (2023, April 15). *What are "Chinese whispers"?* Language Humanities. Retrieved April 23, 2023, from https://www.languagehumanities.org/what-are-chinese-whispers.htm

Data Bridge Market Research. (2022, July 21). Mobile Computing Devices Market Analysis, Segmentation, Size, Share, Trend, Future Demand and Is Expected to Grasp USD 4.70 Billion by 2029. *GlobeNewswire News Room*. https://www.globenewswire.com/en/news-release/2022/07/21/2483787/0/en/Mobile-Computing-Devices-Market-Analysis-Segmentation-Size-Share-Trend-Future-Demand-and-Is-Expected-to-Grasp-USD-4-70-Billion-by-2029.html

Dietz, M., Khan, H., & Rab, I. (2020, August 7). *How do companies create value from digital ecosystems?* McKinsey & Company. Retrieved March 1, 2023, from https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/how-do-companies-create-value-from-digital-ecosystems

Dziuba, A. (2021, August 6). Scaling Software Development Teams: Challenges and Solutions. *Relevant Software.* https://relevant.software/blog/software-development-team-scaling-challenges-and-solutions/

Fentaw, A. E. (2020, June 2). *Cross Platform Mobile Application Development : A comparison study of react native vs flutter*. JYX. Retrieved February 15, 2023, from https://jyx.jyu.fi/handle/123456789/70969

Google. (n.d.). *Patterns for connecting other cloud service providers with Google Cloud.* Google. Retrieved March 2, 2023, from https://cloud.google.com/architecture/patterns-for-connecting-other-csps-with-gcp

Götz, F. M., Stieger, S., & Reips, U.-D. (2017). Users of the main smartphone operating systems (IOS, Android) differ only little in personality. *PLoS ONE*, *12*(5). https://doi.org/10.1371/journal.pone.0176921

Győrödi, R., Adrian, V. G., Zmaranda, D., & Győrödi, C. (2017). A Comparative Study between Applications Developed for Android and iOS. *International Journal of Advanced Computer Science and Applications*, *8*(11), 176–182. Retrieved October 11, 2022, from https://thesai.org/Downloads/Volume8No11/Paper_23-A_Comparative_Study_between_Applications_Developed.pdf

Hartmann, G., Stead, G., & DeGani, A. (2011). Cross-platform mobile development. *Mobile Learning Environment, Cambridge*, *16*(9), 158-171.

Higgins, T. (2022, January 8). *Why Apple's iMessage is winning: Teens dread the green text bubble*. The Wall Street Journal. Retrieved March 2, 2023, from https://www.wsj.com/articles/why-apples-imessage-is-winning-teens-dread-the-green-text-bubble-11641618009

Holzer, A., & Ondrus, J. (2012). *MOBILE APP DEVELOPMENT: NATIVE OR WEB?* https://www.academia.edu/download/38245540/camera-ready-2012-WEB-HolzerOndrus.pdf

Jones, L. (2021, January 29). *Microsoft founder Bill Gates Regrets mistakes with mobile strategy*. WinBuzzer. Retrieved February 28, 2023, from https://winbuzzer.com/2021/01/29/microsoft-founder-bill-gates-says-he-made-mistakes-with-mobile-xcxwbn/

Kim, J., Lee, H., & Lee, J. (2019). Smartphone preferences and brand loyalty: A discrete choice model reflecting the reference point and peer effect. *Journal of Retailing and Consumer Services*, *52*, 101907. https://doi.org/10.1016/j.jretconser.2019.101907

Latour, B. (1996). *On actor-network theory: A few clarifications.* https://www.jstor.org/stable/40878163

*Mobile Fact Sheet.* (2021). Pew Research Center. https://www.pewresearch.org/internet/factsheet/mobile/

*Mobile Operating System Market Share India.* StatCounter Global Stats. (n.d.). Retrieved March 1, 2023, from https://gs.statcounter.com/os-market-share/mobile/india

*Mobile Operating System Market Share United States Of America.* (2022). StatCounter Global Stats. https://gs.statcounter.com/os-market-share/mobile/united-states-of-america

Montecuollo, M. (2014, January 29). *Native or Web-Based? Selecting the Right Approach for Your Mobile App.* UX Magazine. https://uxmag.com/articles/native-or-web-based-selecting-the-right-approach-for-your-mobile-app

Morison, E. (1980). The Uncertain Relation. *Daedalus, 109*(1), 179–184. Retrieved October 11, 2022, from https://www.jstor.org/stable/20024657

Shahzad, F. (2017). Modern and responsive mobile-enabled web applications. *Procedia Computer Science, 110*, 410–415. https://doi.org/10.1016/j.procs.2017.06.105

Turner-McGrievy, G. M., Hales, S. B., Schoffman, D. E., Valafar, H., Brazendale, K., Weaver, R. G., Beets, M. W., Wirth, M. D., Shivappa, N., Mandes, T., Hébert, J. R., Wilcox, S., Hester, A., & McGrievy, M. J. (2016). Choosing between responsive-design websites versus mobile apps for Your Mobile Behavioral Intervention: Presenting four case studies. *Translational Behavioral Medicine, 7*(2), 224–232. https://doi.org/10.1007/s13142-016-0448-y

*What Percentage of Internet Traffic Is Mobile?* (2022, August). Oberlo. https://www.oberlo.com/statistics/mobile-internet-traffic