Version Control & Collaboration In Computer Aided Design

Industrial Design Honours 2017

Harry Anderson

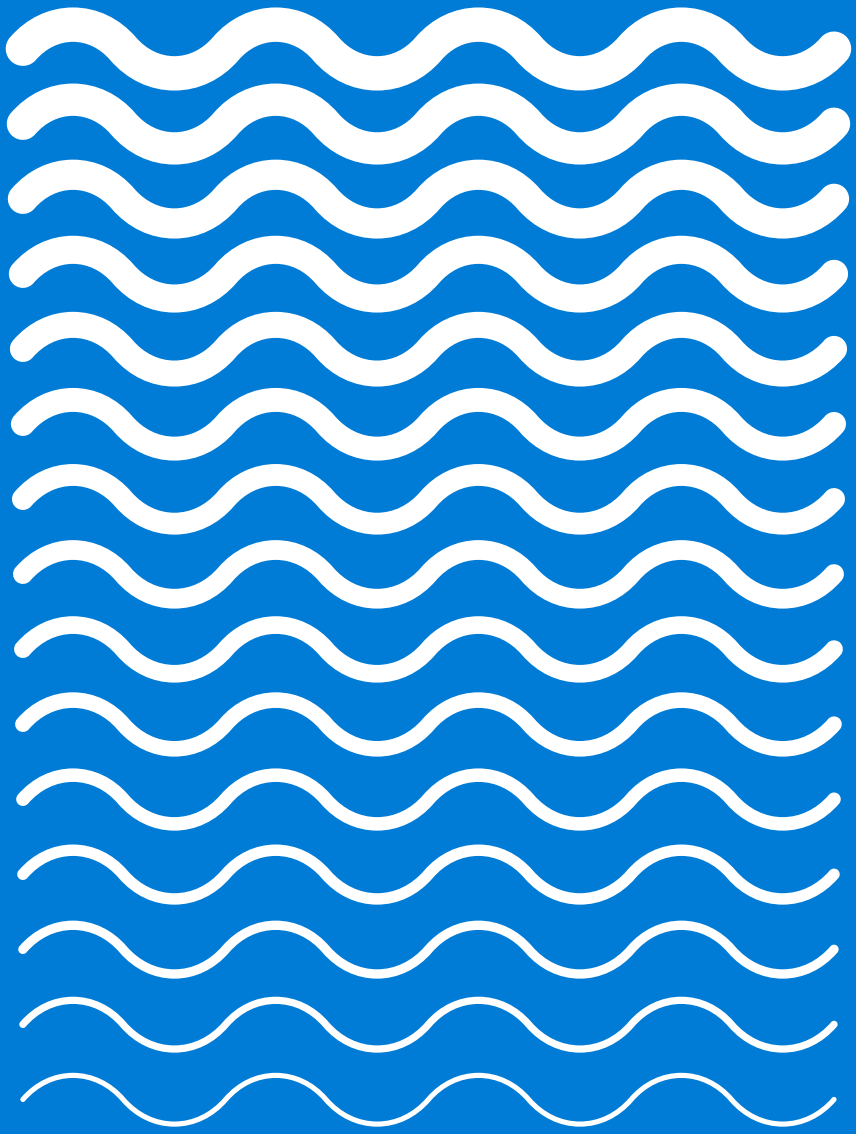
Version Control & Collaboration in Computer Aided Design

Harry Anderson
Bachelor of Industrial Design Honours Thesis
Supervisor: Josh Batty

Email: harryanderson9@gmail.com
Student Email: s3436571@student.rmit.edu.au
Website: hrry.io

RMIT
UNIVERSITY

Contents



Fields of Practice

The proliferation of cloud and mobile technology has led to a paradigm shift in the way in which we use and interact with computers. This essay will explore the contemporary discourses of design within the software development and technology industries. Firstly, current fields of practice will be defined. Namely the fields of, digital product design, user experience design and human computer interaction. Secondly, key topics of discussion relating to the fields of practice such as the relationship between simplicity and functionality and the deepening interactivity of the internet. Finally, the application of these topics will be discussed through relevant case studies such as two similar but different pieces of project management software that recently merged, a micro finance app that allows users to invest their spare change and a cloud based design tool with much greater exercisability than traditional solutions. Digital technologies such as mobile computing and the internet afford designers a new dimension of functionality as well as a new medium for designing products.

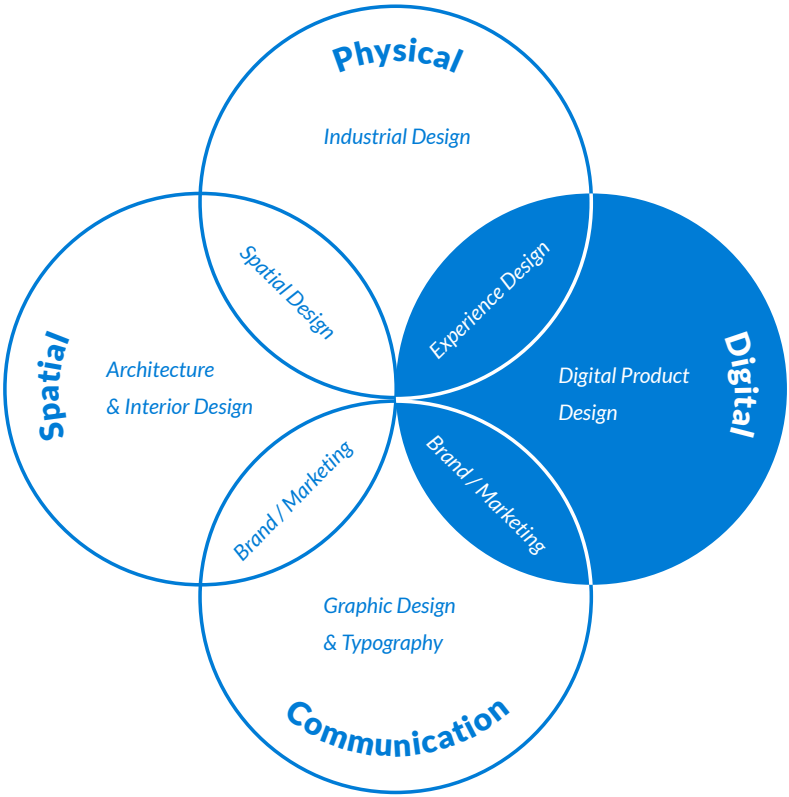
Digital Product Design

Hulbert & Rogers (2007) define Digital Product Design as 'the design of products using embedded digital technologies'. The term 'product design' (used to describe the process of designing physical products) has been synonymous the fields of industrial design & mechanical engineering for over 50 years. However, more recently the term has been adopted widely within software development and technology industries as an umbrella term for design activity.

Maeda (2015) and Kolko (2015) assert the reason for the adoption of product design terminology is the changing role and reasonability of design in software and technology. They state that traditionally, design was equated with styling and was brought in at the end of the product development process to "make the product desirable". Today, design has become involved in every stage of product development from the inception to the final product.

Companies increasingly rely on design for product differentiation and success (Maeda 2015; Kolko 2015). As Maeda reported in 2015 "design is moving from cost for business to an investment".
A Product Designer must consider the needs of the end user alongside the wider strategic considerations of business practice (Kolko, 2015). Similarly, the vice president of design for the online accommodation and hospitality service, Airbnb, compares the role of design, business and engineering to a 3-legged stool – suggesting, "Three elements define a product: the business, the code and the pixels." (Schleifer, 2016).

Designers are also increasingly attracted to digital product design as a field of practice, according to Devay (2015). He suggests that the growing importance of software and technology products has attracted designers from diverse backgrounds as other design fields become highly saturated and loose relevance.



Map of the Field (Kolko, 2017)

User Experience Design

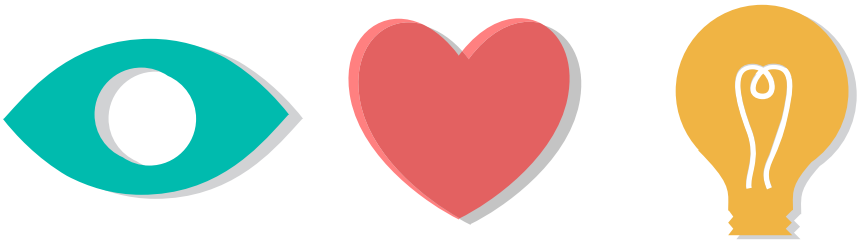
The International standards for ergonomics of human-system interaction defines user experience as "a person's perceptions and responses that result from the use or anticipated use of a product, system or service" (ISO, 2009). User experience design is a multi-disciplinary field that integrates numerous specialisations including user research, interaction design, usability, engineering, marketing and industrial design towards the larger goal of improving the overall experience of users of a given product or service (Norman, Nielsen 2017).

The popularity and demand for User experience design within software and technology industries has increased dramatically in the past 5 years (Maeda, 2015). In part, due to the slowing down in the speed of technological advancement of computational systems. 'Moore's Law', named after Gordon Moore, the co-founder of the prominent micro-chip manufacturer intel, predicted that

"There's no longer any real distinction between business strategy and the design of the user experience"

microprocessors would double in power and half in price every 2 years since the early 1970s. This trend was recently broken when Intel announced in 2015 that they would no longer be able to keep pace with Moore's law as the size of transistors is approaching the atomic level (Simonite, 2016).

Maeda (2015) argues that in the past, software and technology products depended on Moore's law to improve the user experience of their products, explaining that "A better experience was made with a faster CPU or more memory" (Maeda, 2015).



As technological advancement slows down, companies are turning to user experience design to help products feel better and faster in the eyes of the user (Maeda 2015). Maeda also points to the explosion of both users and usage in mobile computing for the increased importance and demand of user experience design in digital products, reporting in 2015 “User experience matters so much because we are experiencing so much” (Maeda 2015).

The demand and popularity of User experience design within business practice also increased dramatically in recent years. Kolko (2015) reports this is due the proliferation of design thinking processes in management consulting and business strategy. The senior vice president of IBM global business services was quoted by Kolko in 2015 arguing, “There’s no longer any real distinction between business strategy and the design of the user experience” (Kolko, 2015).

Human Computer Interaction

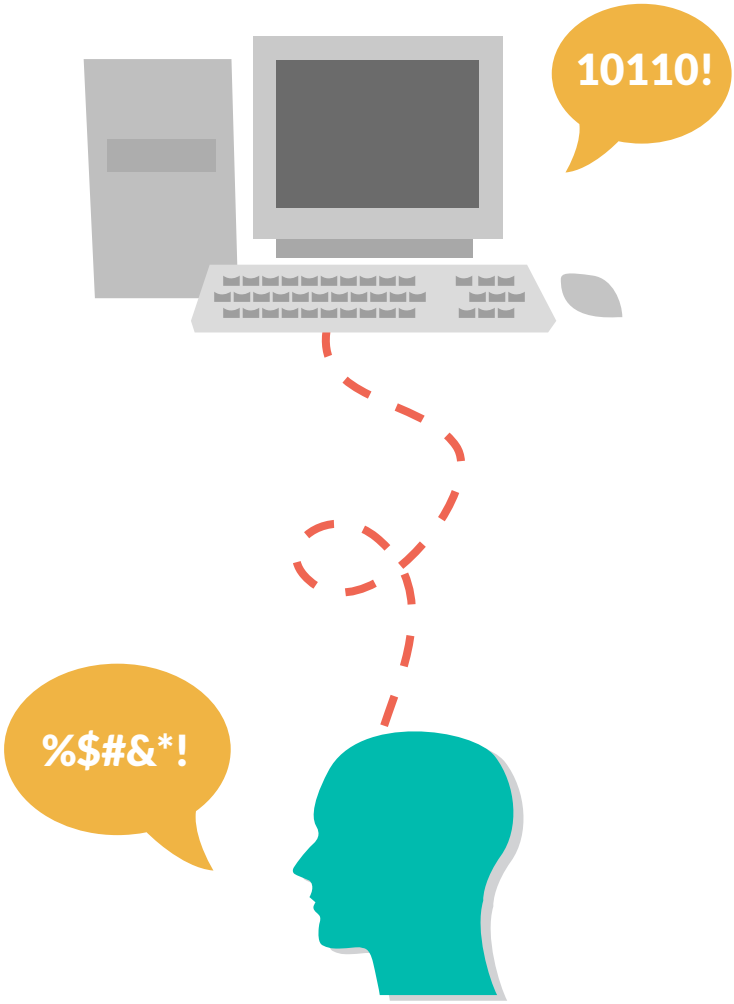
Human Computer Interaction is a research discipline that stands at the intersection of computer science, conative physiology, ergonomics and human factors engineering. Research into human computer interaction began in the late 1980s (Carroll, 2017).

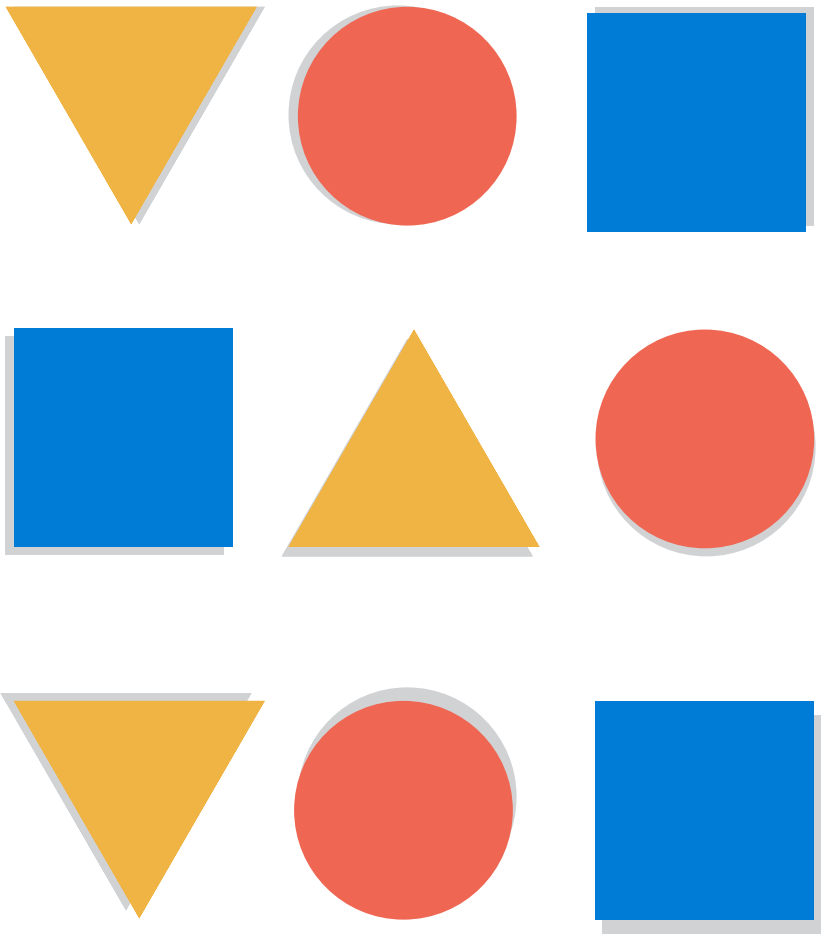
Since then, research in the field of human computer interaction has focused mainly on the interface between human (operator) and computer (system). This is the conduit by which human and computer interact.

One primary objective of human computer interaction research outlined by Card, Newell and Moran (1983) is improving the ease of use of computer systems and optimising the medium by which human and computer communicate. Suchman (1987) suggests within the field of human computer interaction, computers are viewed as 'multi-functional tools' and humans engage in an open dialogue with computers to complete tasks.
This human to computer dialogue is seen as fundamentally different to the human to human dialogue (Suchman, 1987).

Key topics of future research outlined by Sinha, Shahi and Shanka in 2010 include the effects of ubiquitous computing systems on interaction as well as increasing the speed or 'bandwidth' of human computer communication through the development of input methods such as natural speech (Sinha, Shahi, Shanka, 2010)

The Intersection between computer science, ergonomics & Human factors engineering.


10110!
%$#&*!



Simplicity & The Paradox of Functionality

An important point of contemporary discussion within the field of digital product design concerns the relationship between software and innovation. Mike Cannon-Brooks, co-founder of the Australian based software company Atlassian, asserted in 2015 that "every major industry is being disrupted by software", according to Cannon-Brooks, companies have the choice of "becoming a software producer or being disrupted by one" (Clancy, 2015).

However, as Tarasoski (2017), Norman and Nielsen (2017) argue that simplicity & elegance are key to disruptive innovation. Tarasoski (2017) states that products based on disruptive technologies (such as software) are, "typically cheaper, simpler, smaller and more convenient to use". There is a tendency to over deliver on functionality which has the dual effect of increasing the price and complexity of digital products, Christensen (1997) calls this trend 'moving up market'.

Tarasoski (2017) and Christensen (1997) argue that when functionality and price of digital products increases, a vacuum of demand is left for a cheap and simple alternative. Christensen (1997) points to the small business accounting software 'QuickBooks' as one such example, explaining that traditional makers of accounting software had 'overshot' the functionality required for small business accounting. Thus creating an opportunity for QuickBooks to capture 70% of the small business accounting market in two years by offering an alternative product that was not superior in functionality but was simple and convent to use (Christensen, 1997). Similarly, Case (2017) suggests that as technology increases in power, functionality and accessibility, the optimisation of useability and user experience becomes focused on the reduction, simplification and refinement of interactions – either through the reduction of time or attention required to complete tasks.

The Deepening Interactivity of the Internet

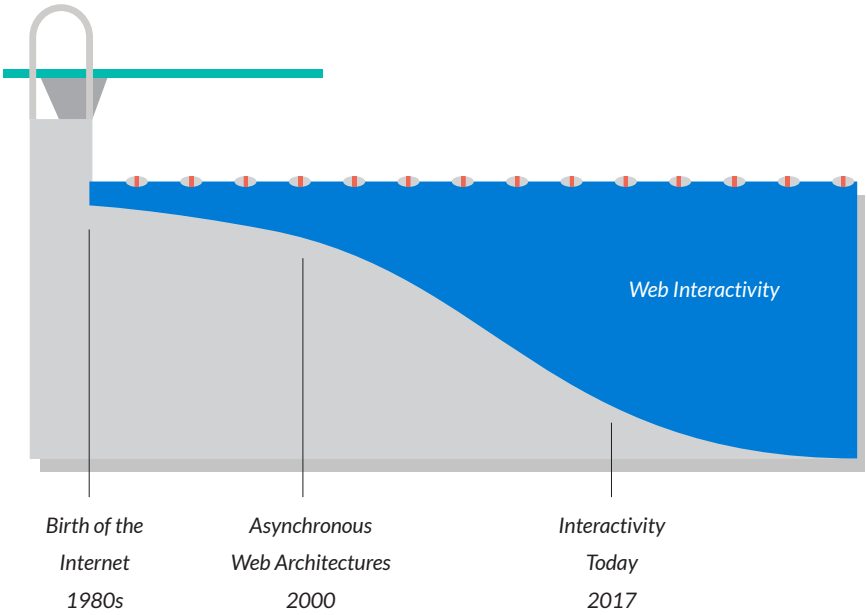
Another important point of contemporary discourse within digital product & user experience design is the deepening interactivity of the internet.
Until the year 2000, web interactions were static - due to the underlying architecture that the internet was originally built on (Garrett, 2005).

Every time a user interacted with a web page, a request had to be made to the server to retrieve the information that the user wanted. If this request was successful, the page then had to be completely reloaded to display the new information. Garrett (2005) suggests, this approach made technical sense but was extremely detrimental to the user experience because “at every step in a task, the user waits some more” (Garrett, 2005).

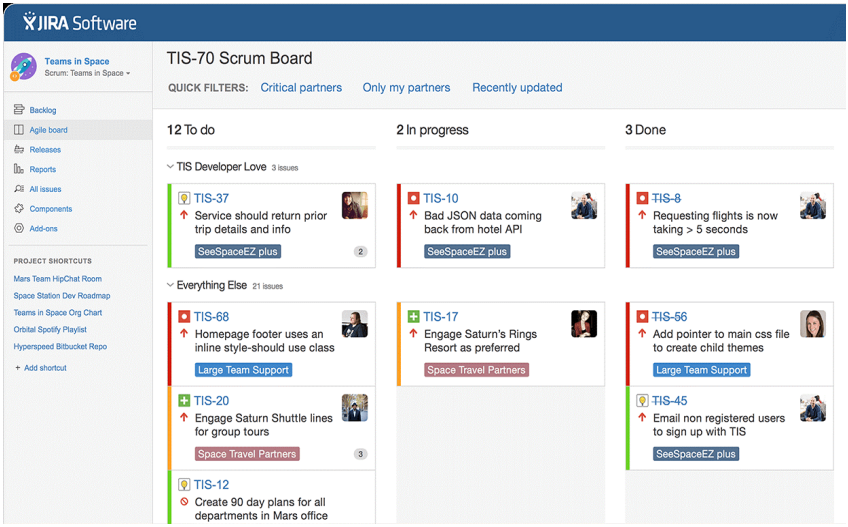
Since the wide spread adoption of ‘asynchronous’ web architectures in the early 2000s - interactivity has developed quickly. Asynchronous technologies such as AJAX allowed for parts of a webpages to be updated to display new information without the need to reload the entire page (Garret, 2005). Today motion graphics and animation are ubiquitous and integral to the design of user interfaces and applications.

Schiefer and Hansen (2015) assert that motion graphics have heightened the user experience by ‘humanizing’ the technology, stating that “Bouncy, wiggly animations give a fun and quirky impression while concise and smooth easing gives the impression of a slick, calculated character.”

With the recent wide spread adoption and support for fast 2D and 3D graphics rendering technology known as 'WebGL' - the future will bring ever deeper interactivity and immersive experiences delivered to users through any modern web browser (Khronos, 2017). One such example is the 'The Curiosity Experience'. A 3D Simulation, created by Nasa's Jet Propulsion laboratory, which allows uses to simulate driving the curiosity rover on the on the Martian surface (Anderson, 2015). The Curiosity Experience is built using WebGL and runs in most Web browsers. According to Kevin Hussey, manager of the Visualization Applications & Development group at NASA's JPL, "anybody with access to the web can now take a journey to mars."(Anderson, 2015)

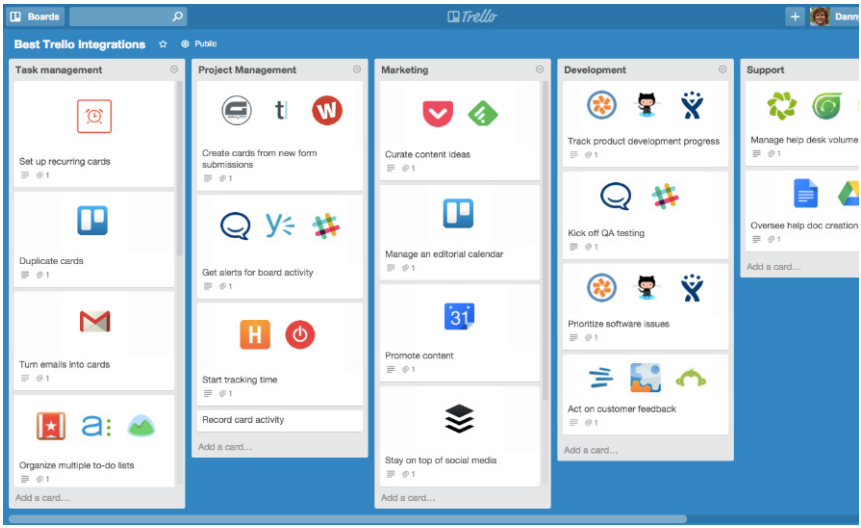


JIRA & Trello



JIRA Project Management Software (Atlassian, 2017)

The paradoxical nature of functionality in digital product design can be illustrated through the example of two similar pieces of project management software. 'Jira' is enterprise level bug tracking and project management application that was developed by the software company Atlassian in 2002. Since its introduction into the market, Jira has found wide spread popularity in large companies due to its powerful functionality, extensive support but expensive monthly subscription. (Atlassian 2017 ; Tarasoski; 2017). 'Trello' is an entry level project management application that was developed by Fog Creek Software in 2011. Trello is designed for small collaborative teams, is very simple learn and is initially free to use (Trello, 2017).



Trello Project Management Software (Trello Inc, 2017)

Tarasoki (2017) asserts that as Atlassian added more and more functionality to Jira that was requested by customers, its complexity and price grew. A phenomenon that Christensen (1997) calls 'moving up market'. Tarasoki (2017) argues that as Jira became more complex and expensive - some users started to seek out a simple and inexpensive alternative, creating a "vacuum for innovation" which Trello filled in 2011. As More and more users abandoned Jira in favour of Trello over subsequent years which led Atlassian acquire Trello in 2017 to protect their core business (Lardinois, 2017). This case study illustrates the power of simplification as a means of disruptive innovation as well as how user experience can influence business strategy and product success.

Acorns Micro Investment App (Acorns Australia, 2017)

Acorns

An example of a digital product that delivers a better user experience through simplification is Acorns. A 'micro-investment' app that allows users to invest their spare change automatically into a diversified stock portfolio (Acorns, 2017). The app is primarily marketed at younger, inexperienced investors.

A better investment experience through simplification & automation.

Tucker (2016) suggests that Acorn removes the traditional barriers to entry for investing such as high fees and complex terminology. Acorns allows users to 'round-up' daily expenses to the nearest dollar, the difference is then added to the Acorn account and invested automatically. Users can choose from 5 stock portfolio options – ranked from conservative to aggressive (Acorns, 2017).

Acorns uses simplicity and automation to dramatically improve the user experience of first time stock investors. By removing the barriers to entry of stock investing, Acorns has tapped into many potential users that had otherwise had been excluded from the market. Tucker (2016) reported that Acorns gained 50,000 new users 3 weeks after launching the app in Australia.

Figma

New web technologies such as WebGL have enabled much greater interactivity online. Applications such as the vector design tool 'Figma' showcase the power of such technology when combined with the accessibility of the internet.

Figma is a cloud based vector editing tool akin to Adobe's 'illustrator' but simplified and focused around the workflow of user interface design (Miller, 2016). The tool is a web application and runs inside any modern web browser thanks to the cross-platform support of WebGL technology. In an article about the design tool,

Constine (2015) suggested that "Figma wants to do for interface design what Google Docs did for text editing". The tool aims to streamline design activity by allowing multiple designers work on the same design file at the same time. The accessibility of the tool also allows for 'non-designers' such as software developers or engineers to access the design files directly - instead of receiving exported assets from the designer. Figma attempts to port other concepts from software development to design work such as version control, shared libraries and integration with other team collaboration tools such as the email replacing instant messaging application, Slack (Constine, 2015).

Hosting design files centrally in the online instead of locally on the designer's computer offers new found freedom for collaboration (Miller, 2016). As the tool is built using WebGL, Figma will run on any modern internet browser which means design files are accessible and editable via any computer connected to the internet.

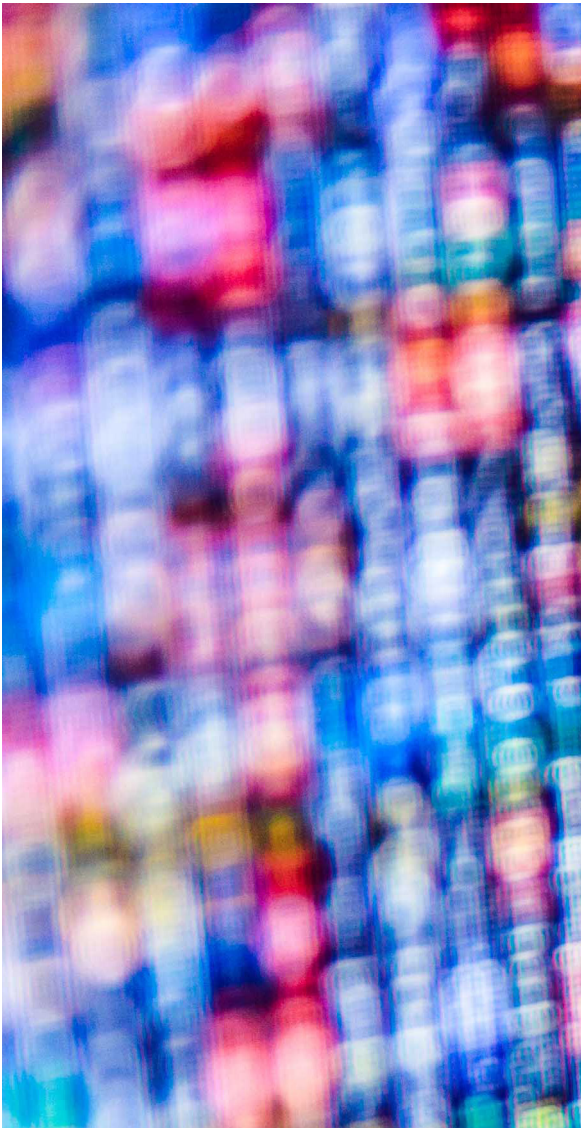
Web Applications such as Figma represent an emerging trend of design tools that turn the web browser and the internet into the next platform for design collaboration.

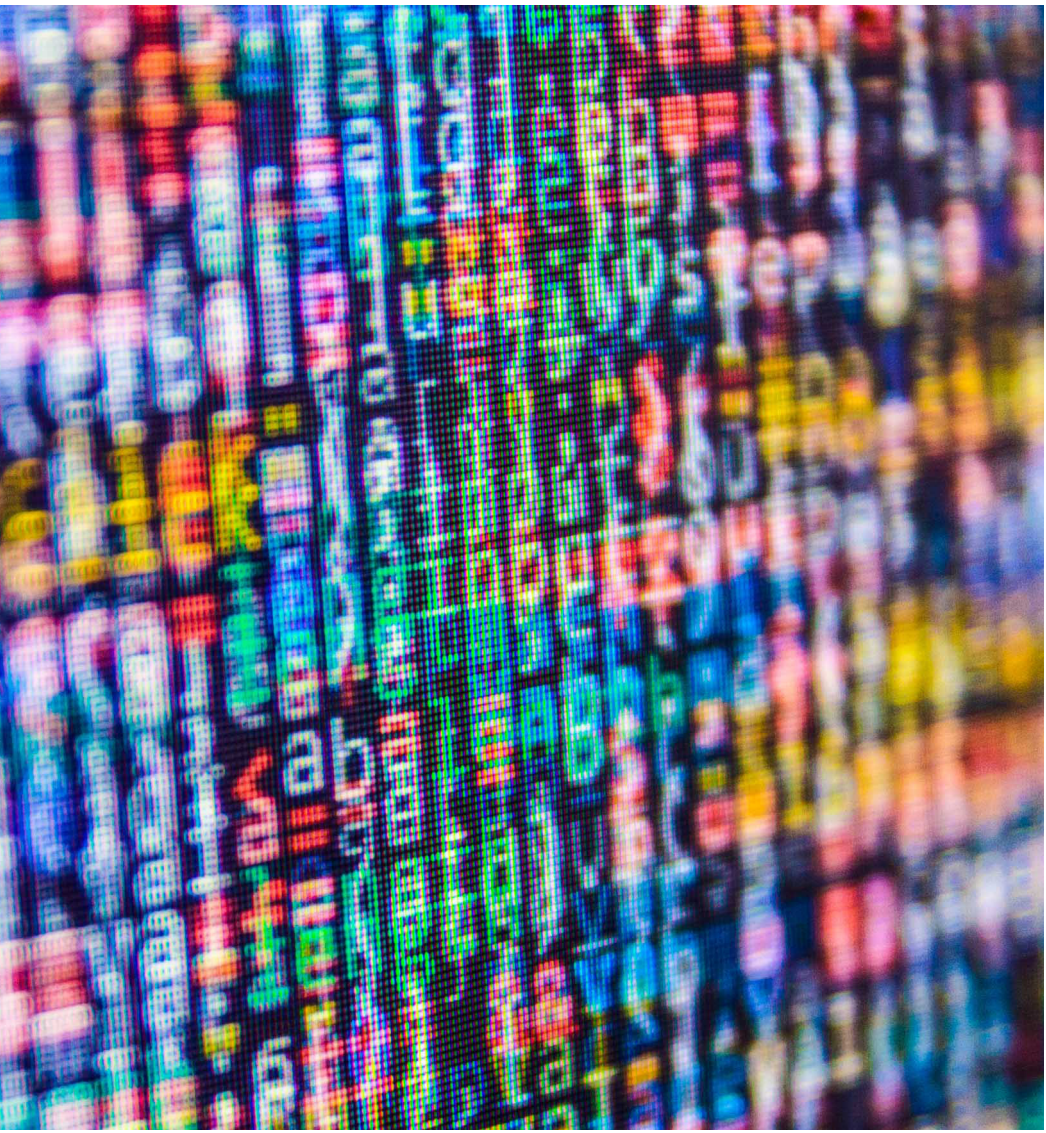
Figma in Use (Source, 2017)

Built using WebGL, Figma will run on any modern internet browser which means design files are accessible and editable via any computer connected to the internet.

Conclusion

Digital technologies such as mobile computing and the internet afford designers a new dimension of functionality as well as a new medium for designing products. Companies are placing higher importance on design of software and technological products as traditional factors of differentiation disappear (Maeda 2015). Designers seek to offer simple and elegant experiences in a world that is oversaturated with information and functionality (Case 2017). The deepening interactivity of the internet is enabling ever more immersive experiences to be delivered to users through traditional tools such as an internet browser – thanks to support for technology such as webGL (Anderson, 2015). As technology becomes increasingly integrated into our lives there is a significant opportunity for the design and development of products that leverage the ever-increasing power and accessibility of computational systems.





Context & Application

The proliferation of cloud and mobile technology has led to a paradigm shift in the way in which we use and interact with computers. This essay will explore the contemporary discourses of design within the software development and technology industries. Firstly, current fields of practice will be defined. Namely the fields of, digital product design, user experience design and human computer interaction. Secondly, key topics of discussion relating to the fields of practice such as the relationship between simplicity and functionality and the deepening interactivity of the internet. Finally, the application of these topics will be discussed through relevant case studies such as two similar but different pieces of project management software that recently merged, a micro finance app that allows users to invest their spare change and a cloud based design tool with much greater exercisability than traditional solutions. Digital technologies such as mobile computing and the internet afford designers a new dimension of functionality as well as a new medium for designing products.

Virtual Work & Collaboration

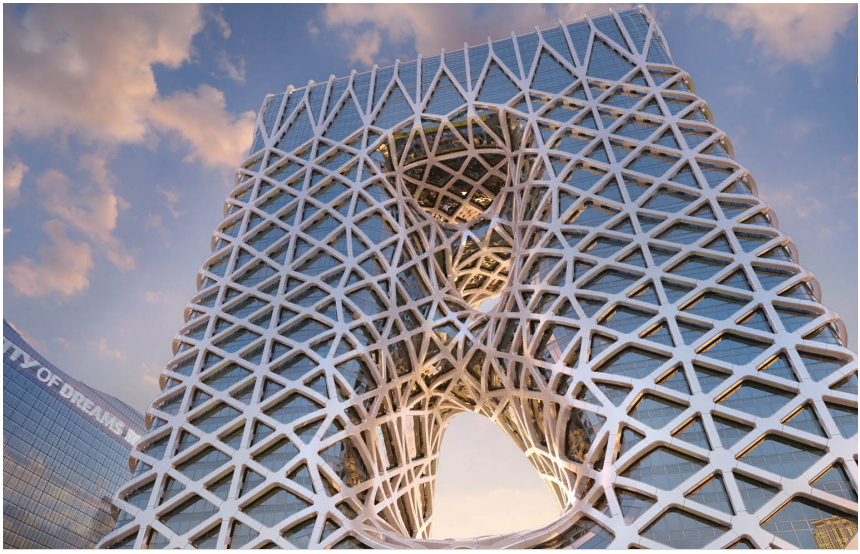
Enabled by new technologies such as mobile computing and the proliferation of the internet, there is a growing trend (at least in developed nations) towards flexible, remote or even entirely virtual work arrangements. Davenport (2005) suggests this trend is particularly prevalent in industries where work tasks are information, experience or knowledge based - known as 'knowledge work'.

A recent survey of 15,000 workers in the US, conducted by the Gallup research centre, found that 43 percent of workers spend at least some time working remotely (Gallup, 2017). Further, the survey found that 31 percent of those surveyed said that they spend 80 to 100 percent of their time working remotely (Gallup, 2017). The report produced by Gallup (2017) went so far as to suggest that flexible, remote and virtual work arrangements are becoming a factor for job selection and employee retention.

New technologies and digital tools are not only facilitating remote working arrangements for knowledge workers in developed nations but are also enabling an unprecedented level of inter-organisational collaboration on a global scale in industries such as architecture, design and the built environment. For example, in a presentation about the design and construction of the 'Morpheus Hotel' in Machau, China. The first high-rise building in the world to be supported by a steel 'exoskeleton'. Associate director of Zaha Hadid Architects, Viviana Muscettoia, stated that "Morpheus Hotel has come to reality through the perseverance of a truly global team of professionals". (ZHA, 2017)

31 % of American workers said they spend 80-100% of their time working remotely in 2017

The engineering, detailing and fabrication of the Hotel's façade was completed by 5 different contractors working in different time zones, which allowed work to continue 24 hours a day (ZHA, 2017). Muscettoia explained that daily video conferencing and cloud based file sharing between Macau, London, Belgium and Shenzhen allowed work to progress at great speed due to constant communication, despite the physical distance (ZHA, 2017). One Industry in which virtual and remote collaboration is standard practice is software development, enabled by digital tools such as version control systems and cloud computing.



Morpheus Hotel Facade (ZHA, 2017)

Version Control in Software Development

Version control systems are a category of software tools that allow for the automatic tracking and management of changes. By monitoring working files of a given project, version control systems track every modification made to the files and make it easy to visualise and manage these changes. Version control also enables multiple 'concurrent' versions of a file to be worked on simultaneously, reviewed side by side and even merged together (Atlassian, 2017).

For instance, one team member may be working on adding a new feature to the project while another team member may be fixing unrelated issues in a different part of the same file.

Using a version control system is considered a must for software development projects and not doing so is regarded as extremely risky and short-sighted due to the inability to recover previous versions (Gratton, 2017).

Atlassian in 2017 asserted that the source code for a software project is "a precious asset whose value must be protected" and that version control systems are an "essential part of modern software team's professional practice". Version control also allows work to progress much faster and collaboratively, due to the ability for multiple people to be working on to the same file at the same time (Atlassian, 2017).

Version control systems are responsible for enabling collaboration on a massive scale in the domain of public software projects, known as 'Open Source' projects – in which anyone can make changes, fix bugs and add features to a project. Anderson (2012) argued that the differentiating factor between successful and unsuccessful open source projects was the forethought of using "a good version control system". The most widely used version control system today is a system known as 'Git' which was developed by the creator of the Linux operating system, Linus Torvalds, in 2005 (Atlassian, 2017).

Git is a 'distributed' version control system in which the entire version history and files of given project, known as a 'repository', are stored centrally as well as locally on each contributor's computer. Git's distributed architecture makes it extremely versatile and is one of the reason for its popularity in commercial projects as well as open source projects (Gratton, 2017).

"Version control Systems are an essential part of modern software teams professional practice."

A service known as 'GitHub' is a popular place for software projects to be stored. Allowing for the working files and version history of a given project to be downloaded in a single package. Any Changes can then be uploaded back to the central repository with relative ease (Atlassian, 2017).

Git and GitHub have had a large impact on the workflow of the software development community and have helped foster a great deal of collaboration within the open source projects by making version management automated and accessible. Anderson (2012) argues that implementing a similar system for "hardware Projects" (design and engineering projects) would foster a similar "community of collaboration" within the product design industry, similar to the effect that version control systems has had on software projects - although Anderson (2012) found there were significant challenges to the idea.

Applying Version Control Practices to Design

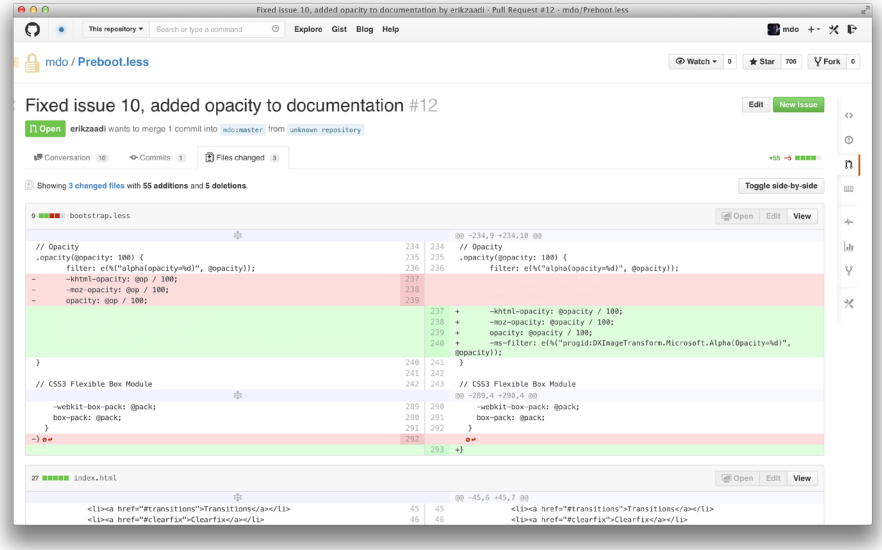
Version control in design is nothing new and is an essential part of managing the development of a design project, especially large and complex projects. Gratton (2017) asserts that while designers do not typically use version control systems such as Git, the key principals of version control are already 'embedded' in how designers work. Saving design files into specific folders with a specific naming conventions like File_001, File_002, File_003 for example.

Files are typically stored locally on a designer's computer or on a file storage systems such as the cloud storage service 'Dropbox' - which allows for teams to access a centralised repository where all design files are kept (Gratton, 2017). However, these version management practices are increasingly seen as outdated in comparison to the version control systems use by software developers and the many benefits they bring to organisation and collaboration.

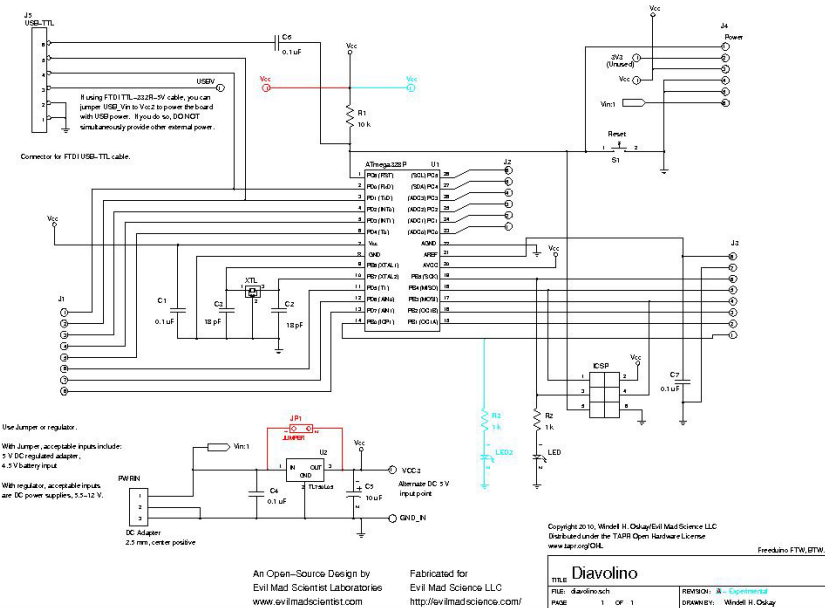
For Instance, the principal designer at 'Adobe' said in 2015 blog post that, "Designers still use shockingly manual and even arcane methods for managing versions" (Vinh, 2015). So why have designers not adopted version control systems such as Git in order to manage versions in the same way software developers do?

Anderson (2012) contends that the problem lies in the nature of design files. Working files for software development, such as source code, are stored in text format - which is easy for a version control system to understand.

If several lines of code are added, deleted or otherwise changed in a text file it is immediately apparent to the version control system and the differences between versions can easily be displayed to the user with corresponding colours.



Comparing Versions on Github (Source, 2017)

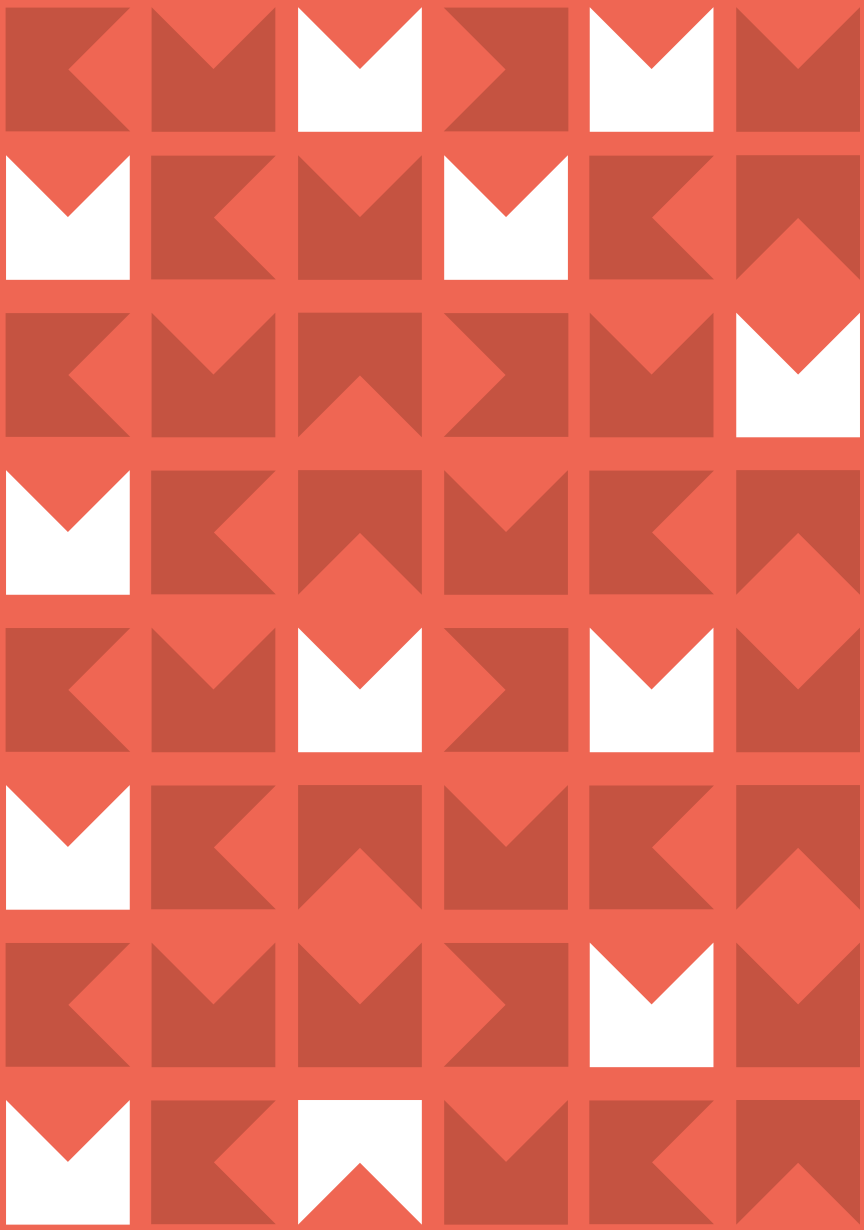


Comparing 'Visual' Differences in PCB Design (Source, 2017)

However, as Anderson (2012) argues, the same is not true for design files - which are stored in binary format and thus differences between versions may not be immediately apparent to a version control system. Design files have no standard format, as some file types are proprietary - meaning they can only be read by specific programs. Typically, design files such as CAD files are exchanged using 'exchange' formats such as STL, DXF, STEP or PDF – however converting CAD files to these formats removes the 'metadata', rendering the design history un-editable (Anderson 2012).

Admittedly, Specialised version control systems do already exist for design files but their use limited to specifically programs, which are typically expensive proprietary solutions such Autodesk's AutoCAD, Fusion 360 or Dassault Systèmes SolidWorks.
One interesting solution to controlling versions in design files that was suggested by Oskay in 2011 is to rely on the graphical nature of design drafting applications. Instead of detecting changes in the underlying code of a design file, differences between versions are detected and displayed visually. Oskay (2011) calls this approach "visual diffs" and he uses the example of electronic PCB design. Different versions of a design are exported as PDF documents and a program detects differences between the files, displaying the modifications to the user in corresponding colours – see fig Y (Oskay, 2011).

While this approach certainly solves the problem of detecting differences in design versions – the example of PCB design lends itself to the 2D format of PDF documents. Oskay's (2011) approach would be problematic for most CAD file formats, in which designs are developed in 3 dimensions. No such system could be found that allows for a user to compare the visual differences between 3D design files before opening the files in CAD applications.



Methods

The proliferation of cloud and mobile technology has led to a paradigm shift in the way in which we use and interact with computers. This essay will explore the contemporary discourses of design within the software development and technology industries. Firstly, current fields of practice will be defined. Namely the fields of, digital product design, user experience design and human computer interaction. Secondly, key topics of discussion relating to the fields of practice such as the relationship between simplicity and functionality and the deepening interactivity of the internet. Finally, the application of these topics will be discussed through relevant case studies such as two similar but different pieces of project management software that recently merged, a micro finance app that allows users to invest their spare change and a cloud based design tool with much greater exercisability than traditional solutions. Digital technologies such as mobile computing and the internet afford designers a new dimension of functionality as well as a new medium for designing products.

Practice Based Research

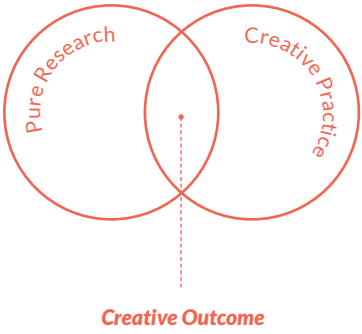
The generation and development of 3 dimensional models using computer aided design software (CAD) is essential to the contemporary practice of industrial design (Tovey, 1989). However, the version control practices used by designers are regarded as extremely manual and outdated when compared with tools used in the software development community (Vinh, 2015).

How can version control practices improve the CAD development process for designers? To answer this question, we must first understand how designers typically manage the CAD development process – both individually and while working in teams. And what the common challenges are that designers face during the development of CAD models. A methodology of practice based research will be used to answer these questions and to establish if there is potential to improve the speed of CAD development as well as ease of communication and collaboration between designers.

The development of 3 dimensional models using computer aided design software is essential to the contemporary practice of industrial design.

Research into the version control practices of designers using CAD software will be conducted using the methodology of practice based research.

(Candy, 2006) defines practice based research as "contribution to knowledge demonstrated through creative outcomes in the form of designs, music, digital media, performance and exhibitions". Research that is practice based has a 'creative artefact' at the centre of the contribution to knowledge (Candy, 2006).



According to Edmond and others, (p. 452, 2007) the field of practice based research is particularly well paired with projects involving new and emerging technologies, stating that "such work often reveals the limitations of existing technologies and opens the door to developing new approaches and techniques" (Edmonds et al., 2007).

For this research project, mixed methods research will be used to gather qualitative and quantitative data which will inform an iterative design process to develop a design outcome. The success of the design outcome will be assessed through a range of qualitative and quantitative criteria generated from the initial research as well as frequent small scale user testing.

Qualitative Data

Qualitative data will be collected from research participants using methods such as interviews, contextual observation and user testing. User's thoughts, feelings and actions will be recorded to gain a rich sense of the user's perspective and to identify any pain points in the user journey.

Two main groups of users will be investigated. First, users of computer aided design software will be interviewed and observed in the context in which they work – these users represent the main user group for research. Qualitative data gathered from this user group will help to gain understanding of the second and third research questions and to validate the research problem.

Second, users of advanced version control and file management systems such as Git and GitHub will be interviewed to gain an expert's perspective on version control. Doing so will help to build understanding of the way in which cutting edge version control tools are currently used by experts outside the context of design and to establish what the most valuable features of such tools are. Qualitative data collected from both user groups will inform the direction, goals and features list of the design outcome.

Method

- User Interview
- Expert Interview
- Contextual Observation

Data Collected

- Direct Quotes
- Users thoughts, feelings & actions
- User Journey

Quanitative Data

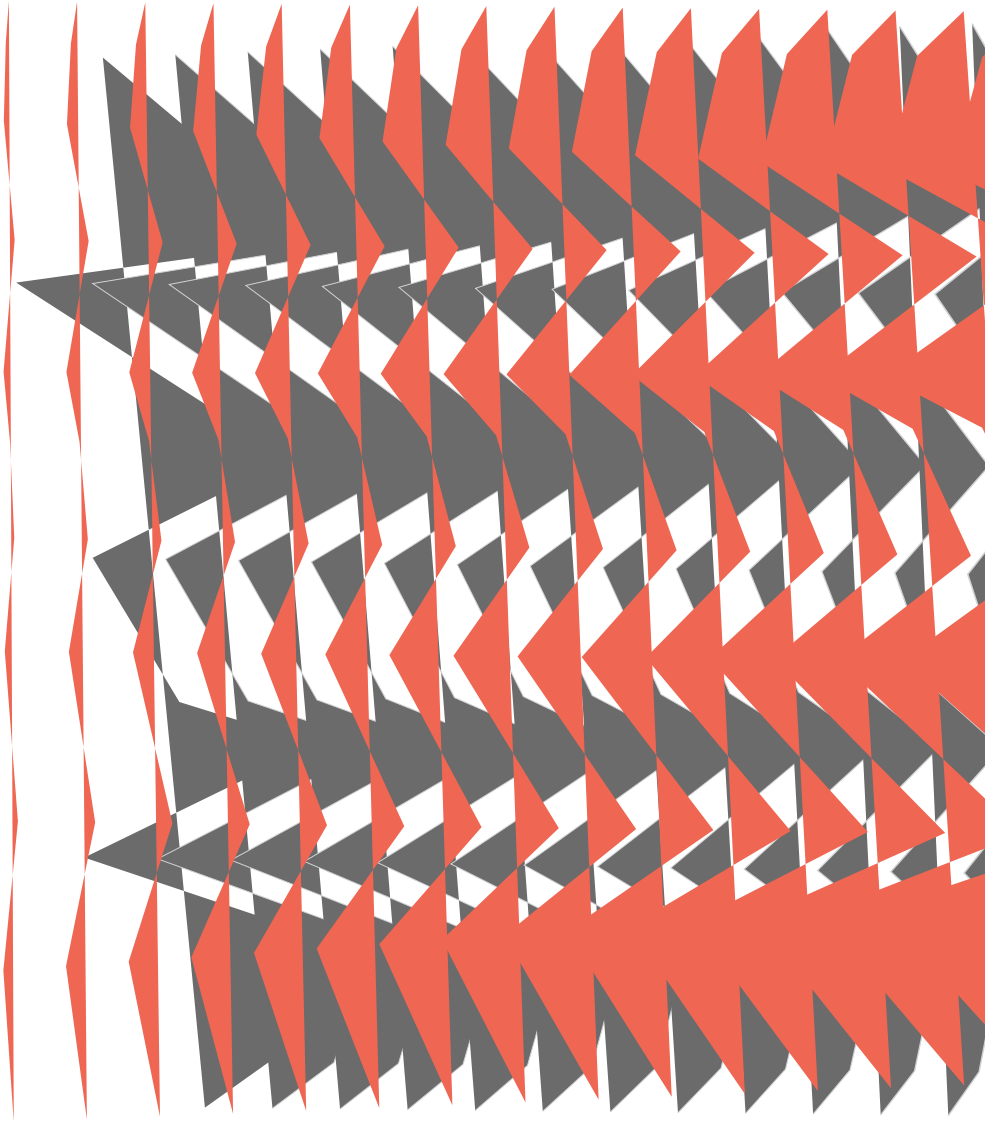
Quantitative Data will also be collected from research participants using methods such as interviews and contextual observation. Quantitative questions such as “how many people are typically working on a CAD model?” or “how many versions would you develop for a given project?” will be used to build understanding of the way in which designers manage CAD development and version control. In this way, discrete and objective data will help to answer the second and third research questions. Quantitative data will also be used to assess the effectiveness of software prototypes through during user testing by making use of user experience ‘benchmarks’ such as the time taken to complete a particular action or the ability to perform an action that was previously not possible before. Doing so will enable any software prototypes to be objectively compared to existing version control practices. Quantitative data will be used to build an objective picture of user behaviour that will inform the direction of the design outcome.

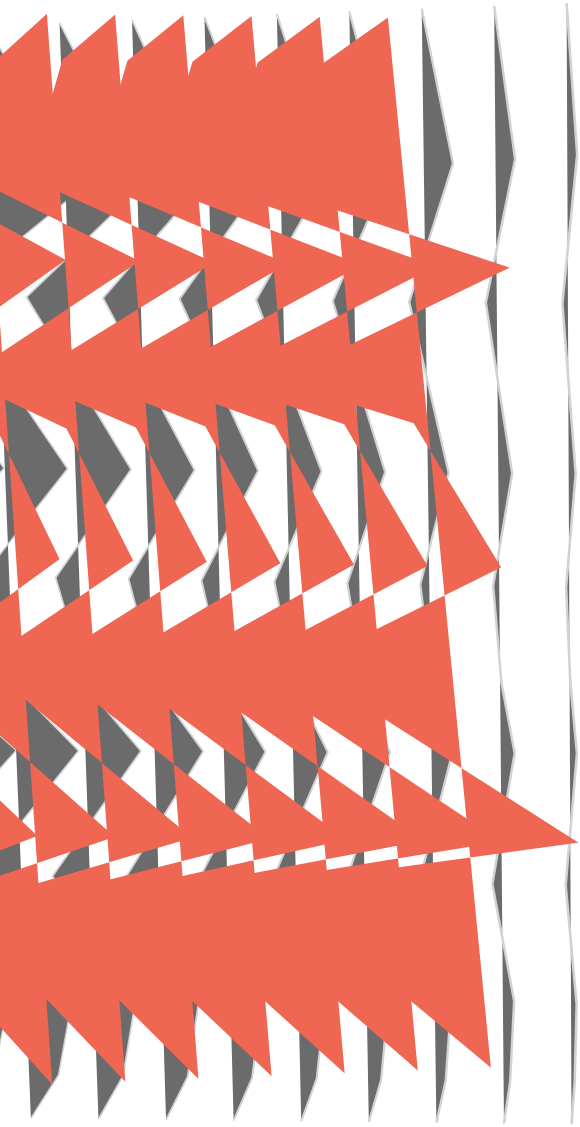
Method

- User Interviews
- Expert Interviews
- Contextual Observation
- User Testing

Data Collected

- Feature Lists
- Objective infomation
- Time measurement





Iterative Design Process

Data gathered from research participants in the initial user research phase of the project will inform the development of a design outcome through iterative design process. Recursive cycles of strategic planning, rapid prototyping and small scale user testing will lead to the development and refinement of successive software prototypes towards the final design outcome.

First, qualitative and quantitative data will be gathered to form bases for a set of goals and features for the first software prototype. Second, a software prototype will be developed using rapid prototyping techniques that meets the goals and requirements set out. Third, small scale user testing is performed to get feedback on the software prototype from users.

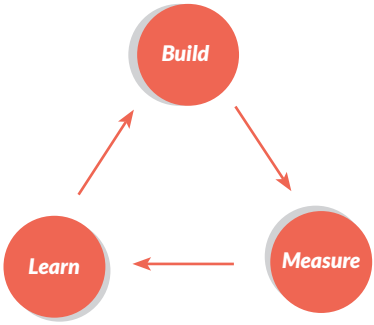
The results of user testing then become the basis for the next cycle of iterative development. Two primary iterative design methods will be used to drive the development of the design outcome.

Lean Development

Lean development is a popular methodology within the software and technology industries for the strategic development of digital products. The Lean development methodology – adapted from lean manufacturing principals - places heightened importance on eliminating waste in software development by only adding features that will add significant value to the users of the product (McLaughlin, 2016).

The aim of lean methodology is to rotate through successive cycles of building product features, measuring their success through qualitative and quantitative data and learning from the results (Ries, 2011). The key principal of the lean methodology is that the results from a given stage become the input for the next stage creating 'feedback loop'. Ries (2011) suggests that the aim of lean development is to accelerate through this feedback loop as fast as possible.

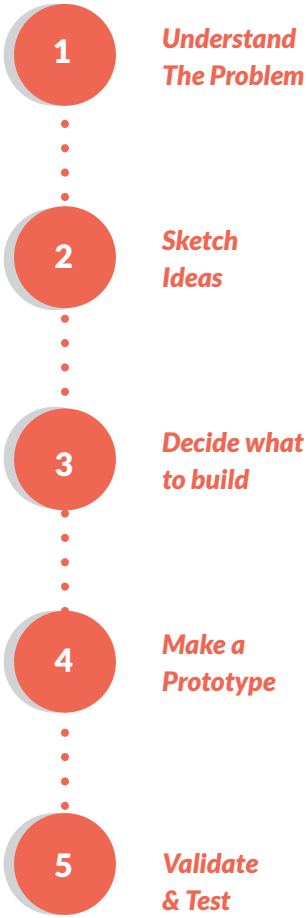
In this way, the development of a product continues while staying closely aligned to the needs to the end user and limiting wasted product development to the bare minimum. Lean development will be used for making strategic decisions and to ensure that prototypes and the final design outcome meet the needs of the end users.



Lean Development Feedback Loop
(Ries 2011)

Design Sprint

The design sprint methodology is a truncated version of the iterative design process that enables development teams to ask and answer design questions in short period of time (typically 5 days). The design sprint methodology was created by Google Ventures, among others, to develop and test product ideas a quickly as possible (GV, 2017). A design sprint consists of a 5-stage process in which the user journey is mapped out to understand the problem, broad ideas are explored, strategic decisions are made about what to build, a rapid prototype is made and tested with a small number of users all within a week (Knapp, 2016). The short time frame allows ideas and assumptions to be tested very quickly, helping to validate the product and insuring features (and the entire product) are valuable to users even in prototype form. Design sprints will be used to rapidly accelerate product iterations from idea to tested prototype in order to learn what is truly valuable to users as a fast as possible.



Evaluation & Validation

Informed by the initial phase of user research, successive software prototypes will be developed through iterative design process. Each iteration will be evaluated against a range of qualitative and quantitative criteria to gauge its effectiveness.

Small scale user testing will also be conducted at the completion of each iteration to gauge progress, get feedback from end users and to decide what to change or add for the next iteration. The evaluation criteria for assessing the success of prototypes is listed here.

Qualitative Evaluation Criteria:

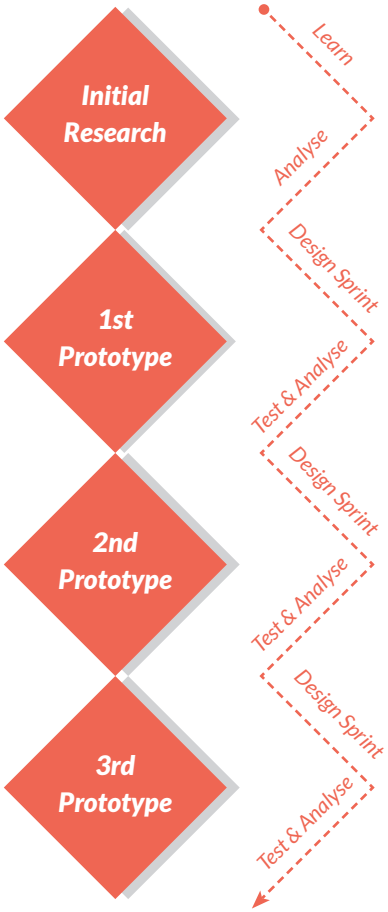
- User reactions & feedback from testing (direct quotes).
- Observation of users during testing (notes).

Quantitative Evaluation Criteria:

- Survey of overall experience (questionnaire).
- Functionality Benchmarking (prototype versus traditional approach).
- Timed action completion benchmarking (prototype versus traditional approach).
- Quantitative information from observation

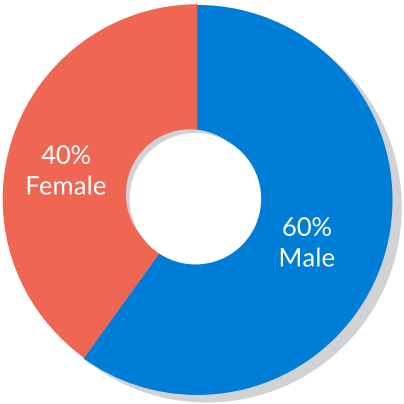
The generation and development of 3 dimensional models using computer aided design software (CAD) is essential to the contemporary practice of industrial design (Tovey, 2016). How can version control practices improve the CAD development process for designers? A methodology of practice based research will be used to answer this question.

First, mixed methods of user research such as interviews and contextual observation will be used to gather qualitative and quantitative data. The data gathered from the initial research phase will inform the design direction of the first software prototype. Next, methods of iterative development such as lean development and design sprints will be used to incrementally improve and test successive prototypes toward the final design. The effectiveness of the software prototypes and the final design outcome will be evaluated against a range of qualitative and qualitative criteria in order gauge its success when compared with traditional approaches of version control.



Quantitative Results

Demographic Information



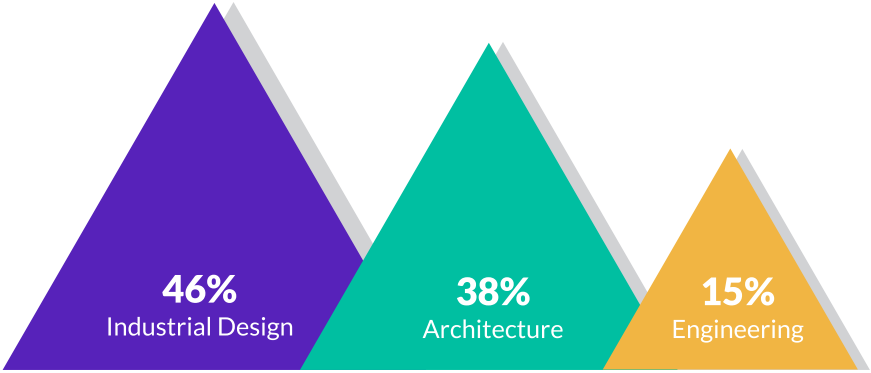
13

Interview Participants

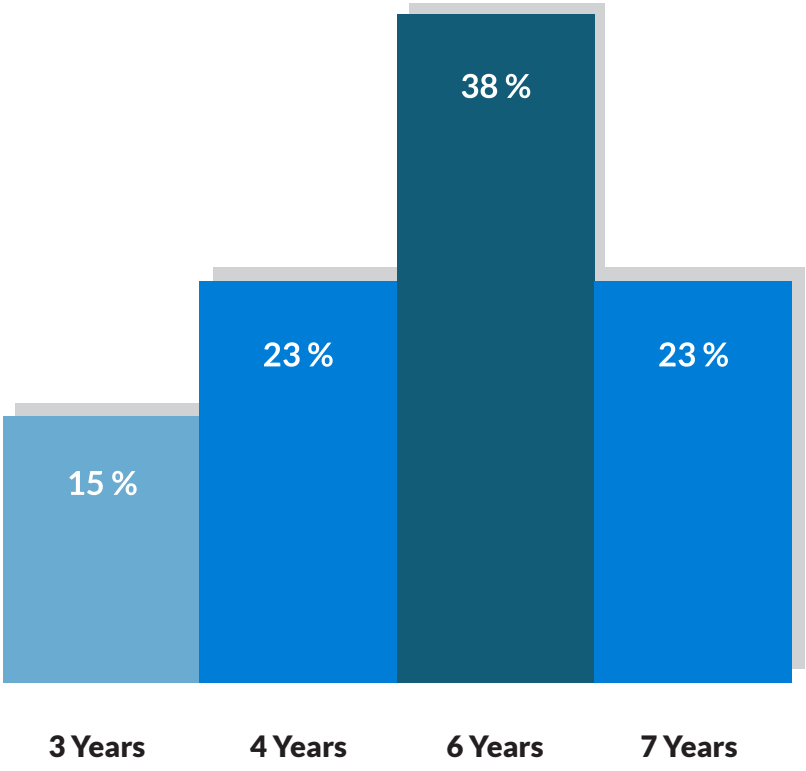
18 - 28

Years of Age

Participants by Discipline



Experience Using CAD Software



Most Common CAD Programs

Working File Types

Solidworks

.SLDPRT ~ part file
.SLDASSEM ~ assembly
.SLDDRW ~ drawing file

Rhino

.3dm ~ Rhino File
.gh ~ Grasshopper File
.gha ~ Grasshopper Plugin / Cluster
.ghx ~ Grasshopper Component

Fusion360

.f3d ~ Fusion360 Archive Format

Evolve

.evo ~ evolve file

AutoCAD

.dwg ~ AutoCAD drawing
.dxf ~ AutoCAD Drawing Exchange

ArchiCAD

.PLN ~ ArchiCAD file

Revit

.Rvt ~ revit Project
.Rte ~ revit template

SketchUp

.SKP ~ sketchup project

Maya

.ma ~ maya project file

3D Studio Max

.3ds max ~ 3ds max project

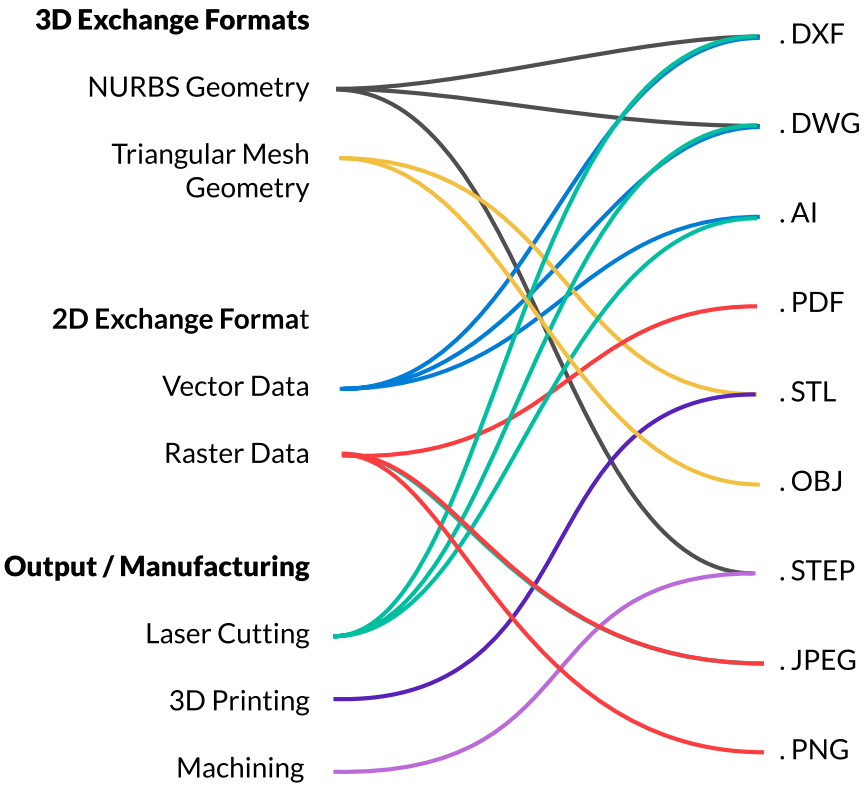
Adobe Suite

.PSD ~ photoshop doc
.AI ~ Illustrator doc
.INDD ~ Indesign Doc

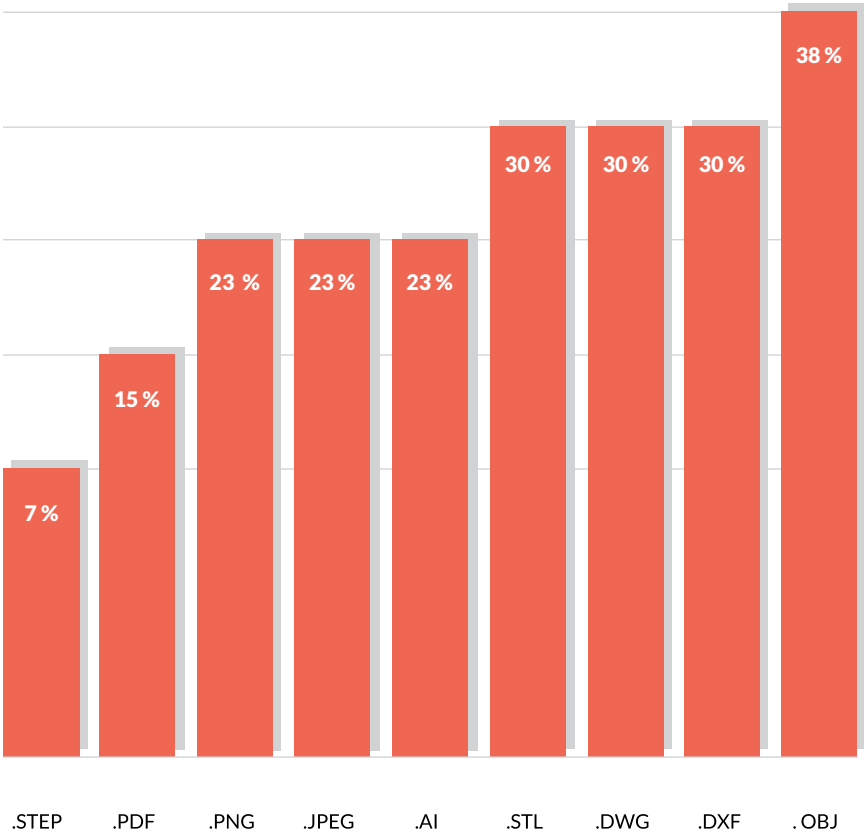
CATIA

.CATPart - Part file

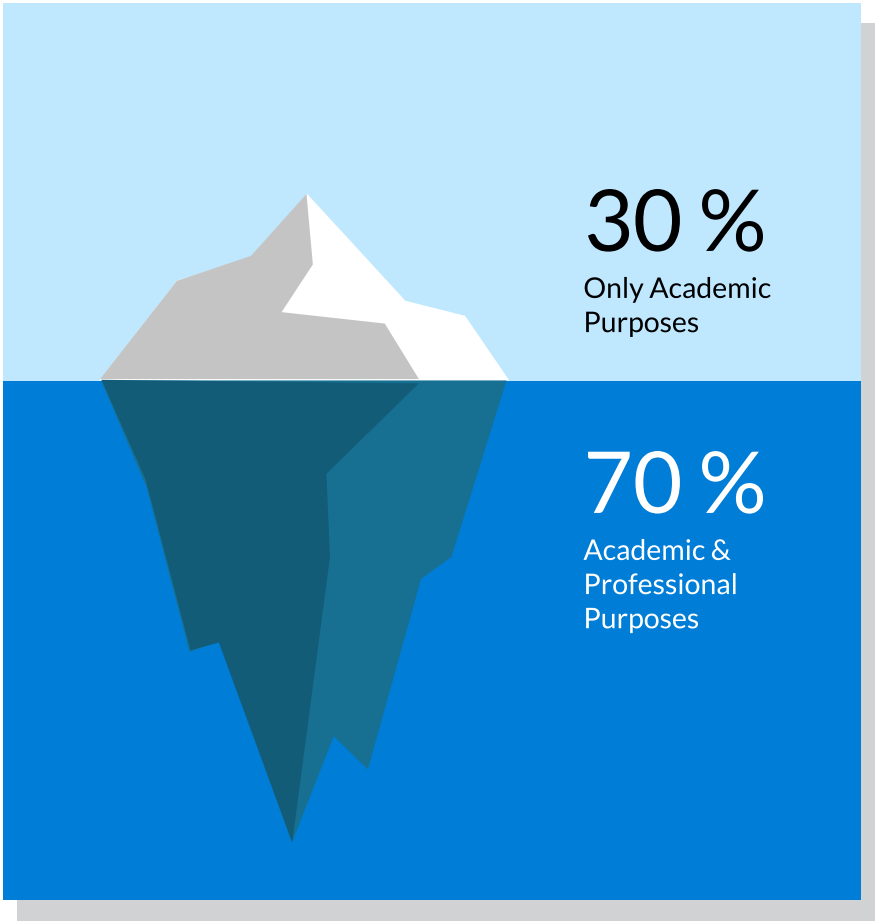
Exchange Files Types



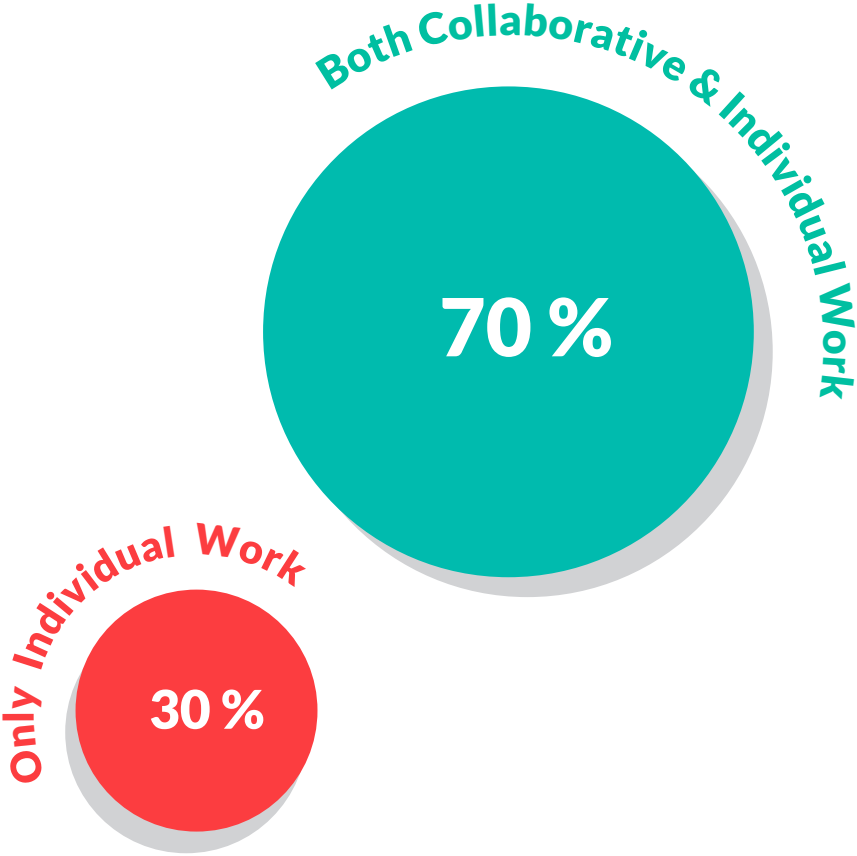
Most Common Exchange Files



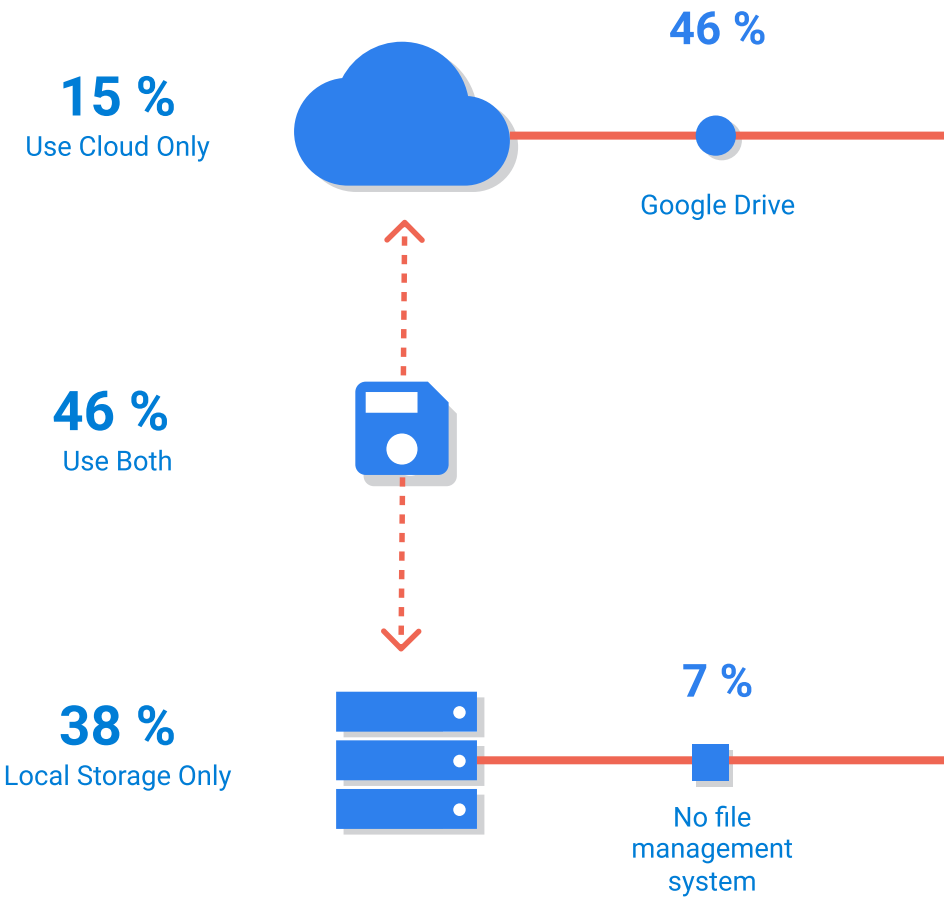
Purpose of CAD Models



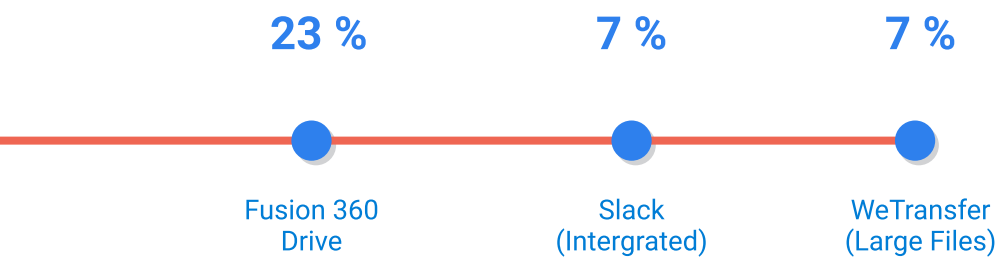
Individual or Collaborative Work



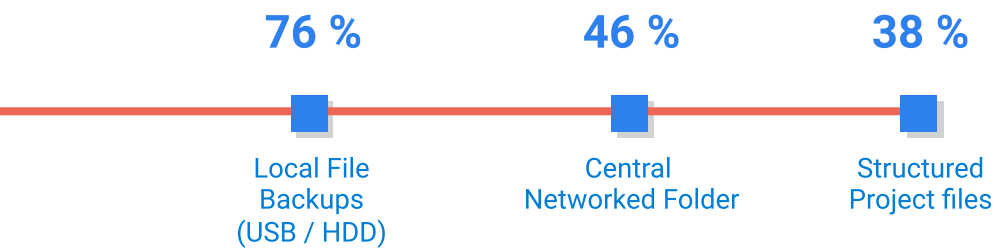
Where Do Users Save their Files?



What Existing Solution Do You Use?



How are local Files Managed?



Chapter 1 References

ACORNS. 2017. Acorns Australia [Online]. Available: https://acornsau.com.au/ [Accessed].

ANDERSON, G. 2015. New Online Exploring Tools Bring NASA's Journey to Mars to New Generation [Online]. Nasa.gov: NASA. Available: https://www.nasa.gov/press-release/new-online-exploring-tools-bring-nasas-journey-to-mars-to-new-generation [Accessed].

AQUINO, A. 2016. What is Product Design. Available: https://medium.com/intro-to-digital-product-design/lecture-1-what-is-product-design-c290bfe799a9#.ff94tud4d.

ATLASSIAN. 2017. JIRA Software [Online]. Available: https://www.atlassian.com/software/jira [Accessed].

CARROLL, J. 2017. Human computer interaction: A brief Introduction. Available: https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/human-computer-interaction-brief-intro.

CASE, A. 2017. Calm Technology O'Reilly Media.

CHRISTENSEN, C. 1997. The innovator's Dilemma, United States, Harvard Business Review Press.

CLANCY, H. 2015. How Atlassian, a favorite with software developers, is redefining business collaboration. Available: http://fortune.com/2015/04/27/atlassian-redefines-collaboration/.

CONSTINE, J. 2015. 23-Year-Old's Design Collaboration Tool Figma Launches With $14M To Fight Adobe. Available: https://techcrunch.com/2015/12/03/figma-vs-goliath/.

DEVAY, P. 2015. What Is Digital Product Design. Node Source [Online]. Available: https://medium.com/@nodesource/what-is-digital-product-design-93caad4e4035#.mq2jzfo48.

DON NORMAN & NIELSEN, J. 2017. Definition of User Experience. Available: https://www.nngroup.com/articles/definition-user-experience/.

GARRETT, J. J. 2005. Ajax: A New Approach to Web Applications. Available: http://adaptivepath.org/ideas/ajax-new-approach-web-applications/.

ISO 2009. Ergonomics of human system interaction Part 210: Human-centered design for interactive systems.

KHRONOS. 2017. WEBGL - OpenGL ES for the Web. Available: https://www.khronos.org/webgl/.

KOLKO, J. 2015. Design thinking comes of Age. Havard business review

LARDINOIS, F. 2017. Atlassian acquires Trello for $425M. Available: https://techcrunch.com/2017/01/09/atlassian-acquires-trello/.

MAEDA, J. 2015. Design in Tech Report 2015. SXSW Conference 2015

MILLER, M. 2016. Figma Is The Online, Collaborative Design Tool UI Designers Have Been Waiting For. Available: https://www.fastcodesign.com/3064096/figma-is-the-online-collaborative-design-tool-ui-designers-have-been-waiting-for.

ROGERS, J. & HULBERT, T. 2007. Digital Product Design. Northumbra University, UK.

SCHIEFER, N. & HANSEN, D. 2015. Driving experiences: Motion Graphics in UI. Available: http://www.effectiveui.com/blog/2015/11/02/driving-the-experience-motion-graphics-for-ui/.

SCHLEIFER, A. 2016. Airbnb.design: Defining product design. Available: http://airbnb.design/defining-product-design/.

SIMONITE, T. 2016. Moore's Law is Dead. Now What? Intelligent Mahchines [Online]. Available: https://www.technologyreview.com/s/601441/moores-law-is-dead-now-what/.

SINHA, G., SHAHI, R. & SHANKAR, M. 2010. Human Computer Interaction Emerging Trends in Engineering and Technology, 1-4.

SUCHMAN, L. 1987. Plans and Situations Actions: The Problem of Human-Machine Communications New York, Cambridge, Cambridge University Press.

TARASOWSKI, M. 2017. How to build the next Trello and sell it for $425 million or more. Available: https://disruptnext.com/how-to-build-the-next-trello-and-sell-it-for-425-million-or-more-589045c9bd64#.txcgbexyr.

TRELLO. 2017. About Trello [Online]. Trello Inc. Available: https://trello.com/about [Accessed].

TUCKER, H. 2016. I tried Acorns, the app that turns your spare change into investments Tech Insider [Online]. Available: https://www.businessinsider.com.au/review-i-tried-acorns-the-app-that-turns-your-spare-change-into-investments-2016-3#AZ12gG18Icg4zrsv.99.

VINH, K. 2015. Design Tools News No. 7. Available: https://www.subtraction.com/2015/09/30/design-tools-news-no-7/.

Chapter 2 References

ANDERSON, C. 2012. Wanted: Version Control for Stuff. Available: https://www.wired.com/2012/05/we-need-version-control-for-real-stuff/.

ARCHTECTS, Z. H. 2017. The Morpheus Hotel: From Design to Production: Live Webinar [Online]. Vimeo. Available: https://vimeo.com/203509846#t=4m35s [Accessed].

ATLASSIAN. 2017. What is Version Control [Online]. Atlassian. Available: https://www.atlassian.com/git/tutorials/what-is-version-control [Accessed].

DAVENPORT, T. 2005. Thinking For A Living: How to Get Better Perfomance and Results From Knowlege Workers., Boston, Havard Business School Press.

GALLUP. 2017. State of the American Workplace [Online]. Gallup Research Center. Available: http://www.gallup.com/reports/199961/state-american-workplace-report-2017.aspx#chapter-199970 [Accessed].

GIT-SCM 2017. Git Basics. Git-scm.

GRATTON, M. 2017. Git repository for designers as you've never seen: Abstract (+ Sketch). Available: https://blog.prototypr.io/git-repository-for-designers-abstract-sketch-9138cf6ab9b1.

OSKAY, W. 2011. Improving open source hardware: Visual diffs. Available: http://www.evilmadscientist.com/2011/improving-open-source-hardware-visual-diffs/.

Chapter 3 References

CANDY, L. 2006. Practice based Research: A Guide. Sydney, Australia.

EDMONDS, E., WEAKLEY, A. & CANDY, L. 2007. The studio as laboratory: Combining creative practice and digital technology research. Human Computer Studies, 452.

GV. 2017. The Design Sprint [Online]. Google Ventures. Available: http://www.gv.com/spring [Accessed].

KNAPP, J. 2016. Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days, Simon & Schuster.

MCLAUGHLIN, M. 2016. What is Agile Development [Online]. Version One. Available: https://www.versionone.com/agile-101/agile-methodologies/ [Accessed].

RIES, E. 2011. The Lean Startup, Crown Business

TOVEY, M. 1989. Drawing and CAD in Industrial Design. Design Studies.

VINH, K. 2015. Design Tools News No. 7. Available: https://www.subtraction.com/2015/09/30/design-tools-news-no-7/.

User Interview Questions

Broad / Introductory Questions

Can you tell me a bit about yourself and your experience with CAD?
Roughly how many years experience do you have working with CAD software?
What is the purpose of the CAD models you create?
When working on a cad model, do you typically work individually or collaboratively in a team?
What Cad Programs do you use?
what file types do you typically use?

Research Question 1: How do designers typically manage the CAD development process, both individually and while working in teams?

Do you store your files locally or in the cloud? If in the cloud what service do you typically use and why?
Can you tell me about a tough or large project your worked on, how did you manage the development of CAD Versions?
How do you decide to save as a new version? What makes it that the right time to save a new version?
If working on a project with other people, how do you normally manage the development - such as division of the work and sharing files?
How do you communicate changes made to specific versions between you and your colleagues?
How many People typically work on the development of CAD files?
How many versions would a typical project have?

Do you use any sort of system - inbuilt or otherwise - to help manage versions?
Who else needs to have access to CAD files you work on? Do you have to prepare additional files for these people?

Research Question 2: What are the common challenges that designers face during the development of CAD models?

What are some common challenges you have faced during CAD development?
What are some common challenges of collaborating on a CAD project with multiple people?
Have you ever lost CAD files and had to redo part or the whole model? What was the reason for loosing the file?
Have you had any problems with managing file versions?

Contextual Observation Questions - To be asked in the users work context

Can you show me where you keep the working files for a particular project?
What naming convention do you typically use and why?
Can you show me a version of a file and how you would make a new version?
Pretend we are working on the same project, can you show me
how you would share a version with me - including communicating any changes?
Can you draw a diagram or give me a metaphor for how you visualise
versions in your mind?

Expert Interview Questions

Can give us a short description of your current role?
How do you typically organise working files for a project?
What types of files do you typically work with?
Who do you typically share files with?
Are there any software tools that you consider essential to your day to day work and productivity?
Are there any software tools that you use to manage or visualise different versions in a particular project?
Do you use a version control system such as Git? If so what do you used it for?
What would you say are best or most used features of your version control system ?
What are some typical challenges you face when working collaboratively on a project?