2024-07-11

Inclusion4EU: Towards a Framework of Inclusive Software Design Processes and Practices

Damian Gordon
Technological University Dublin, Damian.X.Gordon@TUDublin.ie

Andrea Curley
Technological University Dublin, Andrea.F.Curley@TUDublin.ie

Emma Murphy
Technological University Dublin, emma.x.murphy@tudublin.ie

See next page for additional authors

Funder: Science Foundation Ireland

Inclusion4EU: Towards a Framework of Inclusive Software Design Processes and Practices

Damian Gordon[1] Andrea Curley[1] Emma Murphy[1] John Gilligan[1] Anna Becevel[1] Dympna O'Sullivan[1]
[1]Technological University Dublin, Dublin, Ireland.
Damian.X.Gordon@TUDublin.ie

Abstract

This research is based on a trans-European research project aimed at enhancing software design and engineering practices to promote inclusivity, involving stakeholders from academia, industry, and disability advocacy services. To explore the nature and meaning of inclusion, a number of different approaches to inclusion are outlined, including models of equitable design (Accessible Design, Inclusive Design, Universal Design, and Design for All), User Design processes (Co-Production, Co-Creation, User-Centred Design, Co-Design, and Participatory Design), and Software Engineering Methodologies (Linear, Spiral, and Agile). These three categories of models are combined to form a three-dimensional environment in which software development projects can be mapped into to assess their level of inclusion. Finally, some case studies are presented to illustrate this new 3-D space in action.

Introduction

Digital technology is now pervasive, however, not all groups have uniformly benefitted from technological changes and some groups have been left behind or digitally excluded. As part of a new trans-European research project, Inclusion4EU, a group of European stakeholders from academia, industry and disability advocacy services are exploring ways to improve software design and engineering practices to encourage the development of software systems that are more accessible and inclusive. The key objectives in this process are:

1. Through engagement with stakeholder groups, to develop a set of tangible outcomes (real world case studies and reports) on good and bad practices in software design and development;
2. Via a survey of European institutions to understand the current practices, including best practices, challenges and future needs for teaching inclusive software design;
3. Via a series of co-design sessions with participants from marginalized groups across Europe, to create a shared understanding around the needs, capabilities and preferences of older adults and people with disabilities for inclusive technology;
4. The publication of a co-created framework for inclusive software design and development which will include design patterns, guidelines and checklists to maximize technology inclusion;
5. The creation of a European Community of Practice on inclusive software design and development align that will strongly with European Digital Inclusion initiatives.

This paper focuses on Project Activity 1, with a focus on findings related to good design practices. In this abstract we provide a review of models of inclusive design, as well as processes of co-design that focuses on incorporating diverse users into the design process.

Equitable Design Models

There are a range of design models that consider the issue of inclusion, these include models such as Accessible Design, Inclusive Design, Universal Design, and Design for All. Although these terms are

used interchangeably, they represent distinct philosophies of design, with different origins, and each is used more frequently in different disciplines. and they represent a successive widening of the target audiences. They are, however, all focused on design that includes the authentic consideration of people with a wide range of abilities.

Accessible Design

Accessible Design means designing a product or service that can be accessed by anyone, regardless of whether the individual has a disability or not (Armitage, 2016), and the simplest way to build in accessibility is from the beginning of the design process (Kalbag, 2017). There are a variety of potential accessibility issues that a user may have - visual issues, auditory issues, cognitive limitations, limited movement, speech disabilities, neurological limitations and temporary issues (Barrell, 2019). In the early days of the World Wide Web, Berners-Lee (1997) stated that "*it is critical that the Web be usable by anyone, regardless of individual capabilities and disabilities*". Web accessibility is concerned with ensuring that websites, tools and technologies are designed to be usable and accessible for all users, regardless of ability. The Web Accessibility Initiative (WAI) have developed a variety of guidelines to promote web accessibility that are based on four design principles (Brown and Hollier, 2015): *Perceivable*: e.g. provide text alternatives for non-text content, create content that can be presented in different ways; *Operable*: e.g., make all functionality available from a keyboard, help users navigate and find content; *Understandable*: e.g., make text readable and understandable, help users avoid and correct mistakes; and *Robust*: e.g., maximise compatibility with current and future user tools.

Inclusive Design

An inclusive design strategy requires understanding of diversity within the population and responding to the identified diversity with knowledgeable design decisions that addresses the needs of as wide a range of people as possible (Waller *et al*., 2015). It is very important that Inclusive design is incorporated into the overall design process from the initial concept stage, and all decisions throughout the development process should include the users' feedback (Waller *et al*., 2015). Recent international trends towards the integration of disabled people into the mainstream of society, has been reflected in the inclusive design process (Clarkson & Coleman, 2015). Whilst accessibility design is focused on users with disabilities, inclusive design has a much wider focus as it involves all aspects of diversity (Joyce, 2022). Narenthiran, *et al*. (2022) explored the use of mixed methods to understand how users adapted their personal workspaces during the COVID lockdown, to help develop more inclusive workspaces. To achieve this an exploration of the literature was undertaken, followed by a survey, circulated to students and staff at a large university in the UK, with the aim of understanding how people had adapted their home spaces during COVID lockdown and to explore what barriers they continue to face. The key conclusion of this research was that it is important to work with end users to understand their specific needs and identify creative and inclusive solutions.

Universal Design

The term "universal design" was created by American architect Ron Mace in the mid-1980s (Mace, 1985) to describe a new philosophy of design - "*the design of products and environments to be usable by all people, to the greatest extent possible, without the need for adaptation or specialized design*" (Dolph, 2021; Story, *et al*., 1998). Watchorn, *et al*. (2021) developed a systematic review of current literature regarding applications of universal design to built environments. They used the person–environment–occupation (PEO) model as a theoretical framework for the review, which found 33 key peer-reviewed journal articles. Those articles are generally focused more on description, discussion, and commentary rather than empirical approaches; although, a combination of quantitative, qualitative, and mixed methods approaches is employed in many papers. They conclude that including a wider range of perspectives (occupations, social participation, multi-disciplinary and trans-disciplinary collaboration, and multicultural perspectives) in the ongoing discourse around UD would enable the concept to reach its full potential as a medium for social justice.

Design for All

"Design for All" is a similar notion to universal design, but its focus and origins are more closely related to the development of technologies that are usable by all (Burzagli, *et al*., 2009), as opposed to the built environment. It is not intended to be a design approach to develop a single solution for everybody, but instead as a user-centred approach to providing products that can automatically address the possible range of needs, as such is it often characterized as a "Swiss army knife" approach to design (Nordby, 2004). Harper (2007) explored *Design for All* in the context of the World Wide Web, where he argued that it proposes that every web page should be designed so that as many people as possible can access it, regardless of any sensory or cognitive impairments. However, he observed that the concept means different things to different people, and this creates a barrier to full implementation of it. He notes that for some people it is a broad notion that impacts society at large, by making reference to socioeconomics, ethics, and issues of general discrimination, while others see it only as a technological issue and a problem to be solved.

Summary

In this research we consider each of these models as a progression in terms of the scope of the range of people for whom each considers within their design process, with Accessible Design representing the narrowest range of users, and Design for All representing the widest range of people and devices.

Table 1: Comparison of Models of Equitable Design

	Accessible Design	**Inclusive Design**	**Universal Design**	**Design for All**
Original Discipline:	Disability Studies (ref. Armitage, 2016)	Ergonomics (ref. Waller *et al*., 2015).	Architecture (Ref. Mace, 1985)	Computer Science (ref. Burzagli, *et al*., 2009)
Aimed at:	Specifically focused on people with disabilities.	Marginalised Groups (including age, size, and ability).	Marginalised Groups (including age, size, and ability).	Everyone, and a wide range of technologies as well.
Principles:	W3C/WAI Guidelines including WCAG and ARIA.	Inclusive Design Research Centre guidelines.	CAST Principles of Universal Design	All of the design principles from the other models.

User Design Processes

Software development processes that either include users as part of their development, or even consider users as part of the design process, are called "User Design Processes" (Norman, 1986). They typically involve gaining an understanding of the users and their needs by conducting user research which will lead to a series of goals, tasks, preferences, and pain points of the users. Following this a design process will occur, which may either directly involve users, or the outcomes of the previous stage. Next, a development process will occur, again either involving users or the user research, Finally, the system will be evaluated, typically with real users. Some examples are presented below.

Co-Production

Co-Production describes the development of public services and technologies where citizens are involved in the design process (Pestoff, *et al*., 2013). In many cases it can involve citizens not only being consulted, but also being involved in the conception, design, steering, and ongoing management of public services (Bason, 2010). A typical definition of co-production is "*an asset-based approach to public services that enables people providing and people receiving services to share power and responsibility, and to work together in equal, reciprocal and caring relationships*" (CNW, 2023).

Co-Creation

Co-Creation refers to a joint design process that is similar to the Co-Production process outlined above, however instead of focusing on public services, it focuses on businesses and their interactions with customers. In a co-creation scenario, a business will take ideas and other input from their customers, to strengthen the relationship between them (Lopera-Molano and Lopera-Molano, 2020). The benefit of this approach is that it creates networks between not only the businesses and their customers; but others such as: suppliers, partners, and employees (Ramaswamy and Gouillart, 2010).

User-Centred Design

User-Centred Design (UCD) has its roots in the computer science domain, and it advocates the inclusion of user-centred considerations such as usability goals, user characteristics, and usage environment into the design process (Norman, 1986). It recommends including users in the design process when possible, but if not, allows for the use of alternative approaches such as *personas* (Gulliksen, *et al*., 2003). Some common considerations in UCD developments include legibility, readability, understandability and accessibility (Suojanen, *et al*., 2014).

Co-Design

Co-Design refers to design processes where designers incorporate input from non-designers (including customers, researchers, and other stakeholders) into their design. The nature of the collaboration will vary widely from project to project, and the designers and non-designers may not have an equal say in the design outcomes (Zamenopoulos and Alexiou, 2018).

Participatory Design

Participatory Design describes an approach where designers and non-designers (including customers, researchers, and other stakeholders) actively participate together in the design process (Ehn, 1992). It has its roots as a political movement in the 1970s to help form partnerships between labour unions and employers (Spinuzzi, 2005). There is a collection of methodologies that are associated with participatory design, which emphasizes not just consultation, but active, meaningful participation of the non-designers.

Summary

In this research we consider each of these models as a progression in terms of the degree to which users are involved in the development process, and the amount of control they have in the decision-making processes.

Table 2: Comparison of User Design Processes

	Co=Production	**Co-Creation**	**User Centred Design**	**Co-Design**	**Participatory Design**
Original Discipline	Political Science (ref. Pestoff, *et al*., 2013).	Marketing (ref. Lopera-Molano, 2020).	Computer Science (ref. Norman, 1986)	Design Science (ref. Zamenopoulos and Alexiou, 2018).	Political Science (ref. Spinuzzi, 2005)

Participants	Designers are essential, but non-designers are often included also.	Designers are essential, but non-designers are often included also.	Designers are essential, but non-designers are often included also, or personas.	Designers and non-designers (including customers, researchers, and other stakeholders)	Designers and non-designers (including customers, researchers, and other stakeholders)
Locus of control	Public sector, typically, with the designers.	Private sector, typically, with the designers.	Typically, with non-designers, but not mandatory.	With the designers.	Shared between the designers and the non-designers.

Software Engineering Methodologies

Software Engineering Methodologies describe the project plans required to develop a software system, typically with timelines and tasks. Since the 1960s these approaches have been used particularly "*to develop large scale functional business systems*" (Elliott, 2004). These approaches have strong parallels with building architecture (Slayton, 2013), where in both cases the needs of the customer are identified, followed by a designing process, a development process, and a testing process. Below are three seminal methodology types, spanning the history of programming, to help explore the evolution of development approaches.

Linear Models

The earliest models of software development are described as "linear models", this means that the software project is divided into several stages, and each stage is undertaken sequentially. Crucially, in this type of model, the developers are not allowed to revisit a previous stage once it has been completed. Once of the oldest, and most seminal linear models is the "Waterfall Model", presented originally by Winston Royce in his 1970 paper "*Managing the Development of Large Software Systems*". It proposes a seven-stage linear model moving from "Systems Requirements", "Software Requirements", "Analysis", "Program Design", "Coding", Testing" and "Operations". Each stage ends with a "Validation and Verification" process where a check is done to ensure that the activity of the current stage matches the outcomes of the previous stage, and a second check to ensure that the activity of the current stage matches the overall goals of the process. It is important to note that Royce's paper specifically warns against this model being used literally, since the "Testing" stage begins so late in the process, and Royce feels that testing should start once the "Program Design" stage is done.

Iterative Models

A newer series of models for software development are referred to as "iterative models", which means, as before, the software project is divided into several stages, however, in this type of model, the developers are allowed to revisit previous stages as frequently as required. The most notable iterative model is the Spiral Model, as described by Barry Boehm, in his 1998 paper "A Spiral Model of Software Development and Enhancement". It presents a radically different approach to modelling the development process, whereby the final system is developed by producing a series of prototypes, and each prototype feeds into the next generation prototype in an iterative manner, until the final release is created. Within each prototype development there are four stages, outlined below:

1. **Determine Objectives**: This stage considers the aims of the current iteration, and details them as a series of requirements, and it also sets out an initial design for this stage.

2. **Identify and Resolve Risks**: This stage looks at the some of the risks that stem from the selected approach and identifies ways to mitigate or eliminate those risks.
3. **Develop and Test**: This stage is similar to the Waterfall Model or V-Model stages, where the systems is full designed, developed and tested.
4. **Plan the Next Iteration**: This stage looks at what has been developed, and looks at the overall goals of the project and how the next iteration will get us nearer to the overall goals.

Agile Models

The newest series of models for software development are referred to as "agile models", which means, as before, the software project is divided into several stages, however, in then further sub-divided into much smaller tasks that can be completed rapidly (Fowler and Highsmith, 2001). One of the most common agile models uses the analogy of the game of rugby to describe the software development process. The Scrum Framework divides the development process into 2-4 week intervals called "Sprints". The entire set of tasks to build the completed system are put in a list called the "Product Backlog", and the tasks to be done in the current sprint are put in the "Sprint Backlog". Within each Sprint the activities are broken down into 1-2 day tasks, and the team of developers typically have a daily meeting (called a "Daily Scrum") where each developer presents their progress for the day, and their planned tasks to the next 24 hours. When each Sprint is completed, there is a review process ("Sprint Retrospective") to reflect on what went well and what went badly in the completed Sprint. There is also another meeting at the end of each Sprint, called the "Sprint Review" where the progress of the team of developers is presented to all the key stakeholders.

Summary

In this research we consider each of these models as a progression in terms of the degree of autonomy that developers have in terms of the tasks they are doing, additionally each of these models sees a reduction in the granularity of stages in these models.

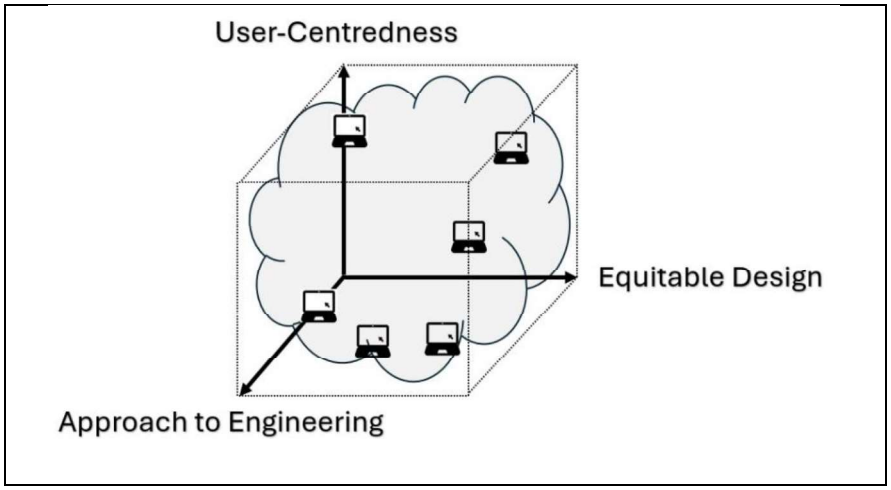
Table 2: Comparison of Models of Software Engineering Methodologies

	Linear Models	Iterative Models	Agile Models
Decade articulated:	1970s	1980s	2000s
Focused on:	Following a clearly articulated methodology.	Reducing potential risks that may occur in the development process.	Getting working software developed that matches the system specification.
Level of Documentation:	All stages and tasks are extensively documented.	All stages and tasks are extensively documented.	All stages and tasks are documented as needed.
Suitable for:	Large software projects with high staff turnover.	Medium software projects with high staff turnover.	Medium software projects with a core team of developers.

The 3-D ED/UC/ATE Design Space

This research proposes a new, three-dimensional design space that allows designers to represent different software development projects in this space to assess the degree to which the users and accessibility are considered as part of the design process, as well as other important inclusion considerations. The three main sections previously discussed are combined into a 3D space where each of the three sections serve as axes in this space. This space is called ED/UC/ATE, standing for: Equitable Design, User Centredness, Approach To Engineering),

This space can be used by software development organisations, either retrospectively, to reflect on the degree to which they have incorporated users and accessibility considerations into their previous projects, and if their trajectory is moving towards more inclusive practices or not. It can be also be used for a current project (or projects), with the recognition that software projects are dynamic by nature, and might move around the design space as the project progresses. In the following section some case studies will be mapped into this space to demonstrate the process.



Picture 1: The ED/UC/ATE Design Space to model the Accessibility of a Software Design Project.

Case Studies

Presented below are a number of accessibility case studies, where the case study is presented followed by a brief description of where this case fits in the 3D ED/UC/ATE space.

Case Study 1: The PLAY-IT Model

Developed by California State University, USA and Leeds Beckett University, UK, this case study focuses on a new approach to making User Experience Design research more inclusive by using an approach called the "Connectivity Model" to more easily include persons with autism in the participation process. User Experience (UX) Design is the process of designing a product or service so that a user has maximum satisfaction in using it, through user research, usability testing and iterative design. Using an inclusive design approach is an important step in achieving that goal, but the inclusion of persons with autism in participating group can sometimes be challenging, potentially due to their lack of sufficient cognitive ability or language skills to participate in the research process in meaningful ways. Further, lecturers and teachers may be reluctant to include such persons due to institutional regulations and ethical concerns. The "Connectivity Model" avoids the requirement for complex ethical clearance by facilitating observations via recorded videos. It analyses user behaviour looking at social, emotional, behavioural, physical and motivational needs, and considers constraints such as ability in the areas of physical, cognitive, and developmental areas.

On the ED/UC/ATE model, this case scores low on the ED scale, as it is focusing on a specific disability, high on the UC scale as the Connectivity Model can be used in many user design scenarios, and high on the ATE scale, because again the Connectivity Model can be used in many engineering methodologies.

Case Study 2: The Ryanair Website

Irish airline Ryanair has a stated aim to reach 225 million passengers by 2026, and a key part of achieving that goal is to improve their digital offerings. A key part of this process is the role of Ryanair Labs, the technology Hub of Ryanair. Ryanair was known in the past for its poor website. Colin O Brien head of

QA at Ryanair Labs identified this as a key reason for growth not breaking through the 80m barriers and this lead to a major re-think about digital strategy. This has lead to renewed emphasis on redeveloping digital services. Lead by Ryanair labs user experience (Ux) is at the heart of this re-design. An extensive process of user feedback is built into development and deployment strategies. Activities include: User Testing, Empathy maps, Contextual Inquiries. Benchmarking, Surveys, and Shadowing. Development involved a 5-stage process Research, Design, Prototype, User Testing and Develop and launch. Users are heavily involved in the Research, testing and Deployment phases. Ryanair continues to grow its digital offering and has added services like a Day of Travel App and Digital Wallet. An important initiative in meeting its goals is its use of a customer panel. By driving a user lead policy Ryainar is on its way to achieving its 225 m passenger goals.

On the ED/UC/ATE model, this case scores high on the ED scale, as it is trying to capture as wide an audience as possible, high on the UC scale as they undertook many user design scenarios, and low on the ATE scale, as the methodology was relatively linear.

Conclusions and Possible Next Steps

This research presented a number of different approaches to inclusion including models of equitable design (Accessible Design, Inclusive Design, Universal Design, and Design for All), User Design processes (Co-Production, Co-Creation, User-Centred Design, Co-Design, and Participatory Design), and Software Engineering Methodologies (Linear, Spiral, and Agile). These three categories of models were combined to form a three-dimensional environment in which software development projects can be mapped into to assess their level of inclusion, this space is called ED/UC/ATE space. Finally some case studies were presented to illustrate this new 3-D space in action.

The next step in this project is to create software tools to allow software developers to model their projects in the ED/UC/ATE space in a dynamic manner. At the start of the project the developers (and others) will be asked a number of questions about the project to develop a baseline for the ED/UC/ATE model, and as the project progresses further questions will be asked to determine whether or not compromises are being made that will diminish the inclusiveness of the project, or on the contrary, if more users are being involved in the project than anticipated, or some other way is used to boost the inclusiveness of the project. It is anticipated that developers will also review existing completed projects to determine their level of inclusiveness, and to reflect upon activities in their organization that can lead to more inclusive software development.

Acknowledgements

This publication/event was co-funded by SFI grant number 22/NCF/OT/11241 as well as the Erasmus+ Programme of the European Union. DISCLAIMER: The European Commission's support for the production of this publication does not constitute an endorsement of the contents, which reflect the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein. This material is based on works supported by the Science Foundation Ireland under Grant No. 19/FFP/6917 and is gratefully acknowledged.

References

1. Armitage, J. (2016). *Bringing Numbers to Life*, The Interaction Design Foundation,
2. Barrell, D. (2019) *Agile Accessibility Explained: A practical guide to sustainable accessible software development*, Amazon Digital Services.
3. Bason, C. (2010) *Leading public sector innovation (Vol. 10)*, Policy Press.
4. Berners-Lee, T., (1997) "World-Wide Computer", Communications of the ACM, (40)2.
5. Boehm, B. (1998) "A Spiral Model of Software Development and Enhancement", *Computer* 21(5), pp. 61-72.

6. Brown, J., & Hollier, S. (2015) "The challenges of Web accessibility: The technical and social aspects of a truly universal Web". First Monday, 20(9).
7. Burzagli, L., Emiliani, P.L. and Gabbanini, F., 2009. "Design for All in action: An example of analysis and implementation". *Expert Systems with Applications*, 36(2), pp.985-994.
8. CNW, Co-production Network for Wales, "What is Co-Production", Available online: https://copronet.wales/home/coproduction/
9. Dolph, E. (2021) "The Developing Definition of Universal Design", *Journal of Accessibility and Design for All*, 11(2), pp.178-194.
10. Ehn, P. (1992) "Scandinavian Design: On Participation and Skill" in *Usability - Turning Technologies into Tools*, pages 96-132, Oxford University Press.
11. Elliott, G., 2004. *Global business information technology: an integrated systems approach*. Pearson Education.
12. Fowler, M. and Highsmith, J., 2001. "The agile manifesto". *Software development*, 9(8), pp.28-35.
13. Gulliksen, J., Göransson, B., Boivie, I., Blomkvist, S., Persson, J. Cajander, Å. (2003) "Key principles for user-centred systems design". B*ehaviour & Info Tech*, 22(6), pp.397-409.
14. Joyce, A. (2022) "Inclusive design". *Nielsen Norman Group*. Available online: https://www. nngroup. com/articles/inclusive-design/
15. Kalbag, L. (2017) *Accessibility for Everyone*. Pearson.
16. Harper, S., 2007. "Is there design-for-all?", *Universal Access in the Information Society*, 6(1), pp.111-113.
17. Lopera-Molano, D. and Lopera-Molano, A.M. (2020) "Designing communities in peace: Participatory action-research approaches embedded in regional education in Colombia", *Gateways: International Journal of Community Research and Engagement*, 13(1), pp.1-16.
18. Mace, R., 1985. "Universal design: Barrier free environments for everyone". Designers West, 33(1).
19. Narenthiran, O.P., Torero, J. and Woodrow, M.,(2022) "Inclusive Design of Workspaces: Mixed Methods Approach to Understanding Users". *Sustainability*, 14(6), p.3337.
20. Nordby, K., 2004. "eAccessibility for all: The role of standardisation in shaping the end-users' tel-eEurope". *TELEKTRONIKK*., 100, pp.4-13.
21. Norman, D.A. (1986), *User-Centered System Design: New Perspectives on Human-Computer Interaction*, CRC Press.
22. Pestoff, V., Brandsen, T. Verschuere, B. (editors), (2013) *New Public Governance, the Third Sector, and Co-Production*, Routledge.
23. Ramaswamy, V. and Gouillart, F.J. (2010) *The Power of Co-creation: Build it with them to boost growth, productivity, and profits*. Simon and Schuster.
24. Royce, W. (1970), "Managing the Development of Large Software Systems", *Proceedings of IEEE WESCON*, 26 (August): 1–9.
25. Slayton, R., 2013. *Arguments that Count: Physics, Computing, and Missile Defense*, 1949-2012. MIT Press.
26. Spinuzzi, C., 2005. "The Methodology of Participatory Design". *Technical Comms*, 52(2), pp.163-174.
27. Story, M.F., Mueller, J.L. and Mace, R.L., 1998. "The universal design file: Designing for people of all ages and abilities", *The Center for Universal Design*, NC State University.
28. Suojanen, T., Koskinen, K., Tuominen, T. (2014) *User-Centered Translation*. Routledge.
29. Takeuchi, H. and Nonaka, I., 1986. "The new new product development game". *Harvard Business Review*, 64(1), pp.137-146.
30. Waller, S., Bradley, M., Hosking, I., Clarkson, P.J. (2015) "Making the Case for Inclusive Design", *Applied Ergonomics*, 46, pp.297-303.
31. Watchorn, V., Hitch, D., Grant, C., Tucker, R., Aedy, K., Ang, S. and Frawley, P., 2021. "An integrated literature review of the current discourse around universal design in the built environment–is occupation the missing link?". *Disability and Rehabilitation*, 43(1), pp.1-12.
32. Zamenopoulos, T. and Alexiou, K. (2018) *Co-design as Collaborative Research*. Bristol University/AHRC Connected Communities Programme.