

University of Economics, Prague

Faculty of Informatics and Statistics



**Using UX Discovery and Design Thinking to build an internal Data Science
knowledgebase for data scientists**
MASTER'S THESIS

Study programme: Applied Informatics (ISM)

Field of study: Information Systems Management

Author: Ahmad Raslan

Supervisor: Ing. PhDr. Antonín Pavlíček, Ph.D.

Prague, December 2019

Declaration

I hereby declare that I am the sole author of the thesis entitled “Using UX Discovery and Design Thinking to build an internal Data Science knowledgebase for data scientists”. I duly identified all citations. The used literature and sources are stated in the attached list of sources.

Prague 1/12/2019

.....

Ahmad Raslan

Acknowledgement

I hereby wish to express my appreciation and gratitude to the supervisor of my thesis Ing. PhDr. Antonín Pavlíček, Ph.D. I would also like to thank my MSD consultant Dr. Michael Wurst for his comments and feedback during the course of this work

Abstract

The primary goal of this work is to apply design thinking approach and User Experience (UX) principles to a real-life use case and create a product that would solve a problem as an end result of following the design thinking process. Design thinking process helps to identify the right problem that needs solving, and then helps to figure out the right solution while UX help in designing the experience of the product assuring that the user interaction with the product works to their best ability to give the user a product that is useful and easy to use. Design thinking has many different models, in this paper different models are evaluated and then the Double Diamond Model is chosen to apply the design thinking process. The real life use case is in the company MSD with the data science team that exists in different locations in the company and the overall problem is that data scientists would need more support specially new hires when it comes to knowledge sharing and knowing the ins and outs of technology and processes. The Double Diamond is followed and UX principles are taken in account while designing the solution and the end result is a data science knowledgebase product called Data Science Portal.

Keywords

User Experience (UX), design thinking process, double diamond model, knowledgebase, Data Science, Data Science Portal, User feedback, User interface (UI), Discovery session.

Content

1. Introduction	3
2. Literature review	5
3. User experience and design thinking	6
3.1. User experience principles	9
3.2. UX vs UI	15
3.3. Bad design use cases	17
3.4. Design thinking process	21
3.5. UX and design thinking models	24
3.6. Double Diamond Model.....	31
4. The Data Science Portal	34
5. Web Technology	37
5.1. Atlassian technologies: Confluence	37
5.2. Atlassian technologies: Jira	40
5.3. HTML, JavaScript, CSS	43
6. Description of the problem and the business need	46
7. Building the Data Science Portal	50
7.1. Project kick-off and team selection	50
7.2. UX discovery workshop to find out relevant content	51
7.3. Article creation process	56
7.4. UI design in confluence environment	57
7.5. User feedback – collection and analysis	58
7.6. Launch of the Data Science Portal	67
8. Data Science Portal success evaluation and current status.....	68
9. Summary.....	70
10. Conclusion	73
11. List of figures	74
12. List of sources	75
13. Appendix	79
13.1. Appendix A: The Data Science Portal screenshots.....	79
13.2. Appendix B: Design iterations	85

1. Introduction

In today's world we truly live in the age of innovation and innovative products. Every year there is at least one new product that haven't existed ever before, yet we adapt to it and cannot imagine our lives without it, that is because the purpose of an innovation is to enhance people's lives and provide true value to those who use said product.

With the level of technological advancement in our world today the number of products entering the market are huge. Everyday there is a new app that offers some sort of service to people, every year there is tens or maybe hundreds of new smart phones and new gadgets and everyone is trying to innovate the next big thing. One might think that innovation is the outcome of technology but that cannot be further from the truth.

Yes, technology is essential to create and innovate a new product regardless of what the product is the technology is there, but at the same time technology is available for everyone, the 3D printing to the ability to process a huge amount of data and run a machine learning models for millions of times in few hours or in some cases few minutes just on our laptops which is something that was impossible just few years ago. The technology is not the key to innovation rather it is a means of implementing innovative ideas and it's a helping hand to better products.

With that said, almost 95 % of all new products fail according to Harvard business school (Claveria, 2019), the number of failed new and innovative products is very high to the point that it makes it not worth going through all the hassles of creating something new or getting into business with a start-up.

There are so many different reasons why a product would fail, a creator might think that his/her idea is gold and will make him/her the new Mark Zuckerberg, or the next Steve Jobs then get hit in the face with reality. The reality of that the idea that a product creator thinks is great has no actual user need. With technology we can create the most advanced and the most innovative product but if there was no real user need, no real problem that the product will solve for the user then the product will be doomed to failure.

From that it is important to understand that the most important aspect of any product innovative or repetitive is the users of the product. Any product should be intended for a group of people aka the users, and a product should provide value or solve a certain problem to its users by satisfying a specific or a group of needs.

User needs are not necessarily always clear or existent, in fact a user need can be solely created by marketing campaigns but nonetheless regardless if the need is real or created by intensive marketing as long as it exists then a product that addresses this need has a fighting chance in succeeding in the market and making profit.

Knowing what users need and what brings value to them to actually make them use your product is not an easy task, all different types of biases and past experiences play a role of making us mistakenly believe that we understand what users want or make us believe that an idea is great and everyone will love it and be willing to actually spend their hard earned money to get the product that consumes that idea.

In order to be able to identify users need and identify users' real problems, and whether an idea for a product would provide the solution two fields of science are perfect for this task. Design thinking and user experience (UX), both of these fields work in harmony to help finding the right problem to solve, then design a solution that users would love to use.

Design thinking can be applied to any field or even to one's personal life to help solve problems but in our context, we will be discussing design thinking from the perspective of creating products.

The main two parts of creating any successful product are: first to actually identify what challenges or problems will be focused on to solve, then second to find and create the solution for those challenges. Putting it in very simple words that the purpose of design thinking and UX, a design thinking approach help us to find the root of the problem, it helps us narrow our focus on the right issue to solve or in other words it enables us to "design the right thing". Once the right problem is identified design thinking helps us to find the right solution or in other words "design things right. Once a solution is identified and validated then technology can come back in the picture and be a factor in productizing the solution idea.

In the design and implementation of the idea into a product that's where UX can be very affective in helping not only creating the product that would address the problem identified by design thinking but UX helps to ensure that the product offer to the user more than its goal. UX design goal is to help the user achieve their goal from the product while having a pleasant experience that would ensure that the user will be willing to use the product again.

The importance of the design sometimes is overshadowed by the functionality of the product but that can easily be proven wrong. If we just think about our past experiences with any sort of products that achieve their purpose but were difficult to use or navigate through, we immediately get back the bad feeling we had with that product and be glad not to have to experience or use those products again. An example would be just looking at most of the university's students information systems, they all achieve their purpose of allowing the students to register subjects and check important information throughout their study, but mostly these systems are horrible in design and a nightmare to use for new students who haven't gone through the process for few semesters over the years. That is the core of UX design it helps ensure that the user while achieving the goal he needs from using a product but without any frustrations or difficulty and provide the user with an easy pleasant experience that would make the user having positive feelings and thoughts about the product.

Now for the purpose of this thesis, the goal is to apply design thinking approach and UX in real life and create a successful product that would succeed of satisfying its target audience of users. Design thinking has many different applicable models, we will evaluate different models of design thinking and choose one to apply to a group of problems in the Pharma company MSD (MSD, 2019) that the data scientists in the company are having and identify the right problem to solve then create the right product that will solve it while applying design thinking and UX approaches.

In this thesis I would like to prove that design thinking is for everyone to capitalize on and benefit from in creating products without having a long experience in the field just by learning the process of design thinking and the principles of UX.

2. Literature review

In the creation of the theoretical part of this thesis various books and online materials have been used. In this chapter these books and resources will be described.

1. What is design thinking

This report published by O'Reilly media explains the basics of design thinking and explains how it can be applied to different type of problems in different stages of product development through real life use cases. The report also gives a brief history of design thinking from the 1960s until today.

2. UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, writers, Designers, and Developers

In this book by Edward Stull, the book explains the principles of UX with enough examples to each principle to allow the reader, who is not necessarily a UX professional, to understand the fundamentals of User Experience and understands the impact that UX has on business and products. The book is easy to read and is the perfect start for those who don't have a prior experience in UX in order to obtain enough knowledge to be able to start the UX discussion and take the first steps to apply it.

3. Tragic design

This book document several use cases of bad design and its implications in real life situations. The book shows how bad design can actually kill users or impact their lives in a very negative way. This book really highlights the importance of design by showing the harm that could be done by a bad one.

4. Design Thinking for Entrepreneurs and Small Businesses: Putting the Power of Design to Work

This book provides the theory of design thinking to be applicable to smaller businesses on a smaller budget and to help design thinking spread around small businesses and not just be exclusive to big corporation. The book describes the process of design thinking from Harvard Design School and its phases and how it works to help the reader learn how to apply design thinking in day to day operations and in creating new products.

5. Design for How People Think

The book in general addresses the cognitive process from different regions of the brain to learn how customers think. In this book the author takes the design thinking and UX from the perspective of the minds of the users not their screens. The reader can learn about the "six mind theory" of user experience and how different minds perceive a single experience differently. Also, one of the most important chapters of this book is chapter 17 that explains the double diamond model for design thinking.

6. Design thinking origin story plus some of the people who made it happen

This article provides the documented history of design thinking plus the timeline of important events that shaped the field of design thinking to what it is today starting from the 1960's from the United States and Scandinavia until recent days. It documents very well all-important figures who played a role in shaping design thinking to what it is today.

3. User experience and design thinking

User experience and design thinking are heavily intertwined together as both of these areas work very well together to help people. Where design thinking tries to find the problem while empathizing with a specific group of people who have a problem and it works to find a solution for it, and user experience focuses on an end to end experience of a user of a certain product and tries to assure that such user will get the best possible experience using that product with ease and simplicity, when used correctly and systematically UX and design thinking can work magic to get the right thing done.

The possibilities of innovation and productization nowadays are endless since the technology is available for almost everyone and companies can make products reach the market quite faster than before but yet we see a lot of products fail and doesn't catch the eye of users. One of the main reasons for this issue is that the user need for such product could be non-existent and the product that was built with the highest kind of technology doesn't really solve a problem that the users have or isn't easy and simple enough for users to be able to benefit from in a straight forward way. That is exactly the role that UX and design thinking play in companies focusing on identifying real problems and finding a way of solving them that would make the user having their problems solved and having a product that not only takes care of his/her pain points but also be simple and easy enough to use and operate. That's why it's not a surprise when we see the biggest companies in the world have invested in UX and design thinking and incorporated it in their business processes in creating products. Companies like Facebook, Amazon, Google, and of course Apple all heavily invest in UX and design thinking therefore it's not a surprise that they are leading the market (Jewel, 2018).

To understand something, we need first of all to understand its history, where did it come from? Who came up with it? And how did it evolve over time?
Let's look first at the history of design thinking and answer those questions about it.

Design thinking history as it was covered in the book what is design thinking (Mailhot, Siminoff, Stanford, & Silverman, 2017), design thinking can be tricky if we did not distinguish between the term design thinking and the process of design thinking as one of those is quite recent and the other goes back to the 60s of the last century. The term itself was made popular in the year 2005 by David Kelly and Hasso Plattner from the design company IDEO and Stanford University when they established the Stanford design school. However, the process of design thinking can be traced back to the 60s and even the 50s of the last century and it wasn't a one centred effort that happened in one location from one single group but rather it was a multiple and unrelated efforts in different places in the world.

In the 50s and 60s in the united states Buckminster Fuller of MIT had his systematic approach to solving problems where he created a team of experts from different fields related to a specific project and those experts are the ones who would be responsible to think of solutions to the main problem that the project is trying to solve, the noticeable aspect of this approach is that Fuller didn't depend on designers but he depended on people who are experts in their fields as they are the ones who understand exactly the problem and also they are the ones who would come up with the best design suited for the product, an approach quite similar to design thinking nowadays.

At the same time on the other side of the world there was the Scandinavian co-operative design with a very different approach from what Fuller had. Here the approach was

democratic and gives the chance to everyone to be involved in the design process, believing that everyone's opinion matters from experts to workers to anybody that might use a system or a product, the designers role was mostly to facilitate and guide the end users to express what features they would like to have in the product and how they would like to use it, this kind of democracy in design that is including the end user in the design process early on is what is called today the incorporation of user research which is a key element of design thinking.

In the 80s Don Norman the vice president of the advanced technology group at Apple came up with the term user centric design, that had all the focus of the design aspect on the user placing the user at the top as the most important element of an information system, this approach insures the focus on the user needs when creating an information system, the user that is not an engineer or a developer, the system must work to their needs and their needs are the most important aspect of the information system. At the same time in the 80's there was an effort from researchers Nigel Cross and Donald Schon to understand what designers did to achieve a state of creative thinking and found out such techniques like brainstorming and sketching play a big role in reaching a state of creative thinking.

Moving into the 90s with establishment of the company named IDEO that took both approaches of Fuller and the Scandinavian co-operative design where they would have their experts but at the same time engage normal users in the design process. Also, in the 90's design thinking started to shift from designing products and software into designing services and solving hard problems.

Reaching the 21st century design thinking have reached the business world and it stopped being exclusive for design firms but rather an important part of business helping and providing the right mindset and tools to tackle big problems and build creative solutions with the main focus on humans. (Szczepanska, 2017)

Much like design thinking **the history of UX** goes way back from the name UX, the term UX or user experience was first used by Donald Norman when he joined apple and became user experience architect in the 90s and became the first person in the world to have his job title as a UX designer, but UX practices dates way back before Norman with evidence that some of these practices date back to Ancient Greece 500 BC (Basuel, 2016).

In the early 1900s Frederick Winslow Taylor the father of what is later known as the Taylorism have conducted a research in the field of interactions between a worker and the tools they use where the result of his research was his book the principle of scientific management tackling inefficiency in workers and how that setup of the tools in workplace and how the design of the tools can lead to eliminating inefficiency in the workplace and optimize the relationship between humans and their tools, although the goals of his research did not focus on the workers themselves and their needs but rather more on having an environment that would get the maximum output out of workers, we can still see clearly aspects of UX in such early times.

In the 1940s Toyota have developed a new production system that was focused on how humans interact with machines with the main value on humans and creating the best working environment for the workers where their feedback is important and counted for and workers are encouraged to provide feedback which is to this day what UX is all about, providing the best possible experience for humans.

Moving on to the 1950s there was one name which have brightly shined in the field of user experience a man by the name of Henry Dreyfuss who was an industrial engineer who have designed great product of that era such as the Hoover vacuum cleaner, and the table top telephone, Dreyfuss wrote the book “Designing for People” which is still very accurate to this day, his philosophy is focused on people interaction with a product that it should make the consumer feel safer, comfortable, and just happy to use the product which indicates the success of the designer. Where if the product has failed to make the user happy and the product is not comfortable to use then the designer has failed. (Basuel, 2016)

By many experts in the art of UX Walt Disney is considered to be the first UX designer from the 1960s for his famous “ Mickey’s 10 commandments” to his team of engineers who he called imagineers, when building Disney’s theme parks(Dickerson, 2013):

- 1- Know your audience: "Don't bore people, talk down to them, or lose them by assuming that they know what you know."
- 2- Wear your guest's shoes: "Insist that designers, staff, and your board members experience your facility as visitors as often as possible."
- 3- Organize the flow of people and ideas: "Use good storytelling techniques; tell good stories not lectures; lay out your exhibit with a clear logic."
- 4- Create a weenie: "Lead visitors from one area to another by creating visual magnets and giving visitors rewards for making the journey."
- 5- Communicate with visual literacy: "Make good use of all the non-verbal ways of communication such as colour, shape, form, texture."
- 6- Avoid overload: "Resist the temptation to tell too much, to have too many objects; don't force people to swallow more than they can digest, try to stimulate and provide guidance to those who want more."
- 7- Tell one story at a time: "If you have a lot of information, divide it into distinct, logical, organized stories; people can absorb and retain information more clearly if the path to the next concept is clear and logical."
- 8- Avoid contradiction: "Clear institutional identity helps give you the competitive edge. [The] public needs to know who you are and what differentiates you from other institutions they may have seen."
- 9- For every ounce of treatment, provide a ton of fun: "How do you woo people from all other temptations? Give people plenty of opportunity to enjoy themselves by emphasizing ways that let people participate in the experience and by making your environment rich and appealing to all senses."
- 10- Keep it up: "Never underestimate the importance of cleanliness and routine maintenance, people expect to get a good show every time, people will comment more on broken and dirty stuff."

Between the 1970s and 1980s UX have evolved even more with the invention of personal computers and the effort from companies like Xerox with their research that led to the invention of the user interface and the mouse, and Apple’s release of the first Macintosh personal computer, then comes 1995 when Donald Norman came up with name user experience, and to this day UX keeps on growing and proving to have more and more value to all different kind of companies worldwide.

3.1. User experience principles

The UX principles described here are from the book UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, writers, Designers, and Developers by Edward Stull (Stull, 2018).

User experience is a unique field where it's not very simple to define and theorize for multiple reasons, it's a roughly new field and there is a lot of talk about it and many ideas and it's not simple to point at one and say that's it, that is what UX is and that how it works. On the other hand, there is a number of principles or guidelines that are not very specific and narrowed down but rather a way of thinking when working in UX.

- UX is unavoidable
- You are not the user
- You compete with everything
- The user is on a journey
- Keep it simple
- Users collect experiences
- Speak the user's language
- Favour the familiar
- Focus on stability, reliability, and security
- Speed is essential
- Usefulness
- There are people in front of the interface

These are essential principles for the mindset of anyone who wants to work, understand, or apply UX in their business life. We will take a deeper look at each one of those principles to have a better understanding of what each one means and how we can apply it.

UX is unavoidable

Whether you are a UX professional, or you work in a company that have UX department or vendor, or simply you are someone who has any kind of device like a smart phone or a computer you are part of UX. User experience is everywhere and it affects you regardless of your involvement or knowledge on it, just simply by using your smart phone you become part of it by being the human that this phone or even just an application was designed for. If you take a look at your favourite devices or applications or even the most websites you visit regularly, you can notice that all of them have one thing in common which is that you can use them comfortably and easily where these mediums provide you with a service you would simply not use them if they were complicated or hard to use. The credit here goes to UX design that made sure that the users can get what they need from it in the simplest easiest way possible.

UX is trend and it is here to stay so the faster the business understands that the better as its unavoidable and a must have when it comes to developing a new product or a service to invest in **UX research** and **UX design** to achieve mainly two very important goals:

- 1- The user goals (your customers)
- 2- Business Objectives

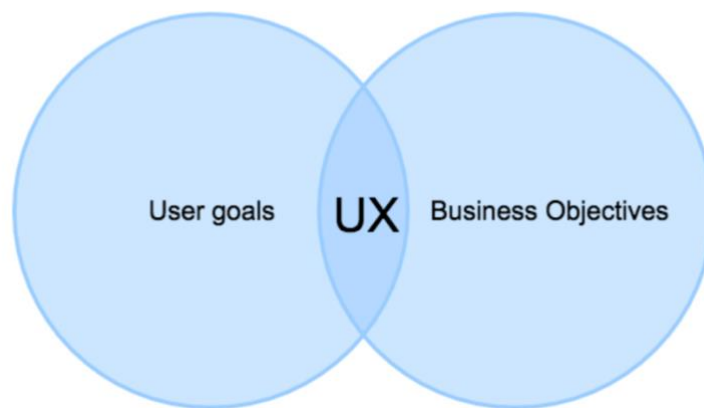


Figure 1: UX connection between business and the user

You are not the user

As simple as this principle sounds, it's crucial for the UX designer to always remember that and quite often UX designers make the mistake of forgetting that they are not the user. UX has two words one is user and the other is experience, it's very important to know the role you play and understand the role the user will play. The user will have an experience and the UX designer is the provider of that experience. As straightforward as these words are in many cases UX designers make the mistake of thinking of themselves as the user specially when designing something that they themselves could use or they have certain amount of knowledge in the area that the product or service provides that they are designing.

As a UX designer you are part of a team and you have a client that requires you to design the experience of their product or service that their target group of users will experience and that can easily introduce a biases into your work without even knowing it, and simply the actual user doesn't share your concerns towards your client or you team who is working with you on this UX project, and therefore it's very important at all time for the UX designers to distinguish themselves from being users to be as objective and unbiased as possible and just focus on the fact that they are the creators of the experience not its users.

You compete with everything

When designing a product or a service we make the mistake of thinking that we only compete with other providers of such product or service, where in fact we are competing with everything else our user might be doing at any given moment, the possibilities are endless and the UX designers goal should be to design an experience that will make the user willingly take part of their time that they could be doing one out of unlimited number of things and experience their product or service. In order for that to happen the experience that the product or service provides must be on a level that makes the user wants to experience it and not feel obligated to do it since there are endless of other experiences and activities the user can be doing other than spending time on your product. A good example that comes to mind is the company HBO who have their online platform HBO GO for streaming movies and series online have had a big success with their famous game of thrones series that gained them a huge number of users yet 16% of their users dropped the subscription after the end of the last season of the show (Yeung, 2019). Indicating that these 16% of users were not happy with the experience they were offered rather they had to stay to be able to see the show they liked and once that was over the experience was not joyful enough to keep them and the company lost them to other activities that they could be doing or to other platforms that offer similar service, especially when we look at their competitors Netflix who in general offer a better UX

than HBO GO (Modrono, 2018). Therefore it's key to understand what and who you are competing with when designing an experience for a certain group of users and the main focus should be on it being joyful enough for the users to embrace it and not give them the feeling that are trying to accommodate it as that wouldn't last long and eventually the users could find another place to get a similar experience.

The user is on a journey

When the user is using an application or visiting a website they are on a journey, this journey has a starting line and has a finishing line a smart UX is the one that keeps an eye on the user during the journey and knows when to offer help or when not to do anything. Three points in the journey of the user that are essentials for the UX designer to plan for are:

- Where the user was
- Where the user is
- Where the user is going

The importance of where the user was is to get a context of the user that would determine the rest of the journey, whether they will continue to the finish line or take the first exit. For example, if a product sells football sportswear the user on the beginning of the journey is probably someone who plays football or planning to play football but if a user that has zero interest in football or its wear, most probably will leave the experience at the nearest exit.

And then where the user is helps the UX designer determine what and if the product should offer help, keeping in mind that too many popups and too many interfering with the experience can have a negative effect and could be distracting for the user. An example would be an application popup asking the user immediately after installation to allow access to the location of the user which is a private information that requires trust between the user and the product which is something that wouldn't be gained at the beginning therefore it can end the experience early in a bad way if the interaction with the user happened at the wrong place at the wrong time.

And lastly where the user is going which heavily depends on the first two points in the user journey. Knowing where the user is going helps the UX designer decide how the journey goes. Knowing how your users behave leads you to making design decisions based on the direction of the users. It's the UX designers' job to guide the user to its goal in his/her experience to get the desired outcome of it.

Keep it simple

When designing the experience for the user in a product or a service it's very important not to drift from the main goal which is to fulfill the need of the user whatever it may be in the context of said product or service. The features are important, but it's necessary to keep an eye on having only the features that add value to the experience and not just complicate it with adding all possible features. It's true that the technology is available and it's possible to add endless features to a product but the higher the complexity the easier it is to lose sight of what matters which is the experience itself. Another point on why to keep things simple is that companies compete for finite numbers of users and if a company would choose to have their products have every possible feature just for the sake of technology and ability rather than what the user needs then that will definitely affect how fast they can productize their creations which would result in competitor companies getting to their target users first. To make sure that the UX designer doesn't lose sight of the experience and to help them not be a victim of complexity there are three methods to understand:

- 1- **Absence:** any feature in the product that wouldn't directly enhance the users' experience and isn't a users' need then it should be absent and removed. Removing distractions for users help them have a better journey in their experience. For example an instant text messaging app can have a feature to let users choose how their texts are encrypted every time they click send but, this feature doesn't enrich the users' experience and on addition it adds another step to click and choose the encryption type and it's just another distraction where the absence of this feature will not affect the users' need but it would make the experience smoother as less unnecessary features means less distractions for the user.
- 2- **Reduction:** analyzing an experience in a product can give the answer to what to be reduced and removed of it. If a website offers three versions with different languages and users mainly use only the English version of the website, then it's necessary to remove the other two versions as they prove no value to the user. Another way of reduction can be reduction to direct the user in certain direction. For example if you want the users to use the live chat option for support then it makes sense to remove the call center phone number or the support email address directing the user to use the feature you want them to keeping in mind that it won't affect their experience negatively.
- 3- **Addition:** is the feature added to the product but in order for it to be a good addition and not a distraction, it should be a feature that decreases the users' tasks and simply adding features that will do something on behalf of the user. It could be something as simple as automatic log in or more complex like predicting what the user is typing next in a search bar, it's all good addition of features as it makes the user do less and that should be the goal of adding features.

Users collect experiences

Users expectations for an experience depends on their previous experiences, thus the past experiences of the user with a similar product effects their future experiences with other products, therefore UX designers need to understand the context of the users' past experiences to be able to design their next one. For example users' have no problem providing their phone numbers in an instant texting app but they will be very hesitant to provide this information in video streaming app and the reason for that is the experiences that users collected from the past have made them accepting providing an information to an app and refusing to provide the same information for a different app. Therefore, the context of the user and their past experiences should be taken into account when designing their next experience.

Speak the users' language

Regardless of the background of the user, UX designers need to assume that users come from different backgrounds and that all of them don't understand technology, business, or marketing strategies and quite frankly they don't care. It's on the designer to take the business goals, the technology, and the marketing and turn it into a meaningful experience for the user in a language that the user understands. It's also important to keep in mind what was mentioned in the previous point about users collecting experiences, as users depend on previous experiences to understand languages and information thus again the context of the user is where we can understand what language to use to talk to the user.

Favour the familiar

A UX designer might be tempted to create something new, something that has never been designed before, designing an innovative experience that would blow the users' mind, but even though innovation is a trend in today's world of advanced technologies, it could be risky

to design a unique experience that was never implemented before. There are multiple reasons for that risk, the first reason would be the fact that 92% to 95% product innovations and startups fail (Nieminen, 2018), which can be scary enough to make any designer not try to be innovative. The second reason is the bias that we as humans tend to have with our own experience. A UX designer while doing his/her job could tend to be affected by his own knowledge and experience assuming that the user would understand or know how to deal with this new innovative experience designed which could be very far from reality. Keeping in mind that the user doesn't know what you know and don't have a similar experience as you do is a key aspect in designing for the user and specially with an innovative UX design since it is not something that the user have seen before or experienced a similar product in their past, therefore, with innovative designs the risk is quite high if the UX designer didn't work in an unbiased way keeping in mind that what matters is what the user needs more than how innovative the experience is. another reason for failed innovations in general is that people tend to not accept change and prefer what is familiar to them specially when it comes to technological adoption as the theory of diffusion of innovation explains (LaMorte, 2019) that the early adopters who would be willing to try a new technology without seeing the evidence that it is what they need are only 13.5% of users(see Figure 2), and for a product or a service to be successful it needs to be accepted and used by at least the early Majority, those users that would need to see an evidence that the innovation works and beneficial for them and the early majority of users are 34% and to reach to them several iterations on the design might need to happen first based on the feedback from early adopters.



Figure 2: Diffusion of innovation theory (LaMorte, 2019)

With that said, innovation is not bad and every familiar experience we have now was at some point an innovation, therefore a UX designer should be careful with innovative designs and take into account the resistance of unfamiliar experiences by the users.

Focus on Stability, Reliability, and Security

the importance of these three components in a product is that it can make or break the users' trust with the product and eventually losing them to a competitor. No one will keep coming back to a website or an application that keeps crashing, or a website or an application that is frequently unavailable without a valid reason that the user understands, or a website or an application that the user feels it is not secure enough.

The **stability** of a product is not noticeable until something wrong happens, a product is stable as long as there are no bugs that interrupts it from performing what it was designed to perform, a bug is not a problem until it appears and the cost of a bug can differ based on when it was discovered. A bug discovered in the design phase would cost 5 times less than a bug a bug discovered in production (Soni, 2019).

It's important to note here that bug fixing isn't rare or isn't something that happens every once in a while. According to a research done in Cambridge University, software developers spend 50% of their time on bug fixing, that is 50% of the capacity and work time of a developer wasted on fixing bugs, therefore it isn't surprising to learn that the same study found that the annual cost of bug fixing worldwide is \$312 billion dollars (Cambridge, 2013).

Taking in account during the design what possible bugs might surface from the experience that is being designed helps to avoid them before they occur. At the end of the day no website or application would run interrupted, errorless, and bugless forever and that's where reliability plays a big role in the experience of the user.

Reliability is basically how long a software, or any digital medium runs uninterrupted. No software runs uninterrupted at a 100% of the time therefore whether its 99% or 99.9 there still will be a significant amount of down time that the user would experience. The UX designer role here is to design how the user will deal with such event. The first important point is to be the first to know when there is a downtime, and that can be easily accomplished having a monitoring system on your product. Then, make sure that part of the user experience is to convey the knowledge on the user in regards to what is going on, a user would be less frustrated buy an error message that explains the situation more than an error message that simply reads "Error" or "Something went wrong", and a frustrated user would lose his trust in a product very fast and would abandon it and look to get the same experience elsewhere.

Security is also a very important aspect of any digital product, even though no matter how secure the product is hackers, spammers, and data thieves still find new ways to hit the security of systems. But the role that the UX designer plays in security is important in the perspective of making the user feel safe using their product. The first step to more secure system is to ask for less information, nobody can steal data that doesn't exist therefore the product should only ask for necessary information from the user and skip all other information, for example if calling or texting the user is not part of the experience then there should not be a field to ask the user to insert their phone number hence less information more security. The second step to make a user feel safe is the design itself, the accuracy of text on each page, as systems that have misspells and language errors seem fishy and a user could think of it as a scam that he/she needs to run away from. A third step to making the user feel that your system is secure is assuring error messages that looks professional and well-designed which would make the user feel that this product is used widely thus it makes them feel safer to use it. For example a dialog box that informs the user about the password requirements upon signing up with enough information to let the user know how strong his/her password is while typing it would make them feel less fearful of signing up and lead them to experience more and continue on their journey in the product.

Speed is essential

The UX designer should take in consideration when designing a users' experience the speed of their journey from the starting point to the finishing point which he/she should reflect to the developers when they write the code. Nothing frustrates the users more than a slow experience. A user who experiences a slow product would hesitate to comeback to it especially that the memory of the speed is usually slower than what it actually was. Determining how much the user anticipates he/she should spend on a task is what the speed should be, and the operation is considered fast if it takes less time that what the user have had anticipated.

Another aspect of speed that must be considered in the experience design is how long it takes in terms of users' tasks for the user to accomplish their goal on the product, which means considering less options for the user to choose from and less distractions while they get from point A to point B. According to Hick Heyman's law (UX, 2019) the time it takes to make a decision increases with the number and complicity of choices, therefore a UX designer should try to limit the number of choices a user needs to make to the needed ones only and discard all unnecessary ones that do not add to their experience in pursue of increasing the speed thus having less frustrated users.

Usefulness

At the end of the day all products are designed to satisfy the users' needs, each product has a specific need that it must satisfy so it becomes useful. Users' needs are not always meaningful and true, users' needs could be entirely made up by the power of advertising. A good example of that is the cigarettes advertising in the early decades of the 20th century, no one needs to smoke but with big marketing campaigns a need was created and companies were competing for customers to smoke their brand, therefore the UX designer should not make a judgement on how important or true the need of a user is. As long as there is a need the UX designer should focus on creating an experience that fulfills this need. With that said a product could be useful and provides the needs of its users' yet the experience is not satisfying, and users' needs are not enough anymore, considering the vast range of options a user can be doing with their time that they chose to spend on your product. Therefore, a complete experience combines users' needs with satisfying experience. And in order to do so three factors that should be added on top of achieving the purpose of the product are:

- Providing useful information to the user.
- Enhancing the capabilities around the users' need
- Keeping in mind the entertainment aspect of the whole experience.

Coming back to an example mentioned earlier comparing Netflix and HBO GO, both of these platforms fulfill the users' need of watching entertaining TV shows and movies, but if we take a look at the video player of both platforms we can see that Netflix provides more information without the user requiring to do any tasks, we can see that the capabilities of the Netflix video player are more enhanced as it offers the users with the option to fast-forward and back-forward button with specified amount of seconds in the video while HBO GO doesn't.

There are people in front of the interface

Lastly the final principle is for the UX designer to not forget that behind the screen where their product is there are people with their own lives and emotions. It's good to remember that you are designing for them, the users are human beings who have decided to give you part of their time and energy so it's a must to try and give them the best experience possible.

3.2. UX vs UI

After studying the principles of UX and understanding them, a logical question one might have would be that isn't UX (User Experience) just another term for UI (User Interface)? The short answer is No. But it is not that simple, both fields have many commonalities and more often than not they work together in harmony to deliver the best looking and working interface that would make the user happy but there are differences between the two fields and definitely they are not the same thing.

We will take a look at each field individually defining each one and explain what it is that each field is concerned with from the aspects of design and research then we will look at the main differences between them and how they complete each other when creating a digital product that users interact with through an interface.

User Experience definition is the singular and accumulated experiences that occur for a user as a consequence of him/her interacting with an object at a given moment in a given context. (Araz, 2018)

Given that UX is concerned with how the interface looks like but more importantly the main focus of it is the overall experience of the user, the interactions of the user with every element on the screen, the flow of the journey in general whether it would make sense to the users or it would leave them baffled and confused about what is their next task should be. Therefore, UX's main focus is the process of the product and how the user will interact with it while still maintaining a good look and feel about the whole interface. A simple example would be that the focus of a UX designer will be more on where clicking a button will take the user and is that the right action at that moment in the journey that the user will be pleased to seeing this button at that moment in that context to let them continue their journey in the product, more than what the color and size of the button is.

On the other side of the coin, the **User Interface definition** is the graphical layout of an application or a website (Planet, 2019). It's how every element on the screen that the user interacts with looks like. From buttons, entry fields, to pictures, or any other element on the screen.

UI is concerned with how a page in an application looks like. How all the elements design on a page correspond to each other from a visual perspective. The UI designer is concerned with choosing the right colors and shapes of all the elements and also the location of them on the page that would make sense to the user to make them feel comfortable interacting with it in an easy way without any confusion. A simple example is if a user is on a page where they need to sign in, it's the UI designer's job to choose where the fields for the user name and password will be, their size and color, the size, color, and location of the sign in button in a way that makes the user feel comfortable and knows what needs to be done without any confusion what every field or button is for.

Differences between UX and UI

From studying the UX Planet article that explains the difference between UX and UI (Planet, 2019)

- Overall difference: the main difference between UX and UI is the aspect of which each field focuses on. UI focuses on visual aspect and the look and feel of an application or website, while UX focuses on the process of it and the user's interaction with each element and how it enriches their journey and gets them to their goal of using said application or website.
- Difference in design:
 - A UI designer works on designing the page from a visual aspect and choose the right colors, shapes, and location of each element on the page.
 - A UX designer works on designing the journey of the user in the page and how he/she interacts with every element on the page.

- Difference in research:
 - A UI researcher's job is to research similar applications that have been done before to find out what works and what doesn't from a graphical design point of view in order not to make a mistake of designing an element that users wouldn't recognize its purpose in order to keep in mind users' expectations of how things should look like.
 - A UX researcher's job is to research the context of the users expected to be using their application or website and research all similar ones from the perspective of users' expectations of how users should interact with the application or website they are designing.
- Difference in medium: another big difference between UX and UI is that UX is applicable to any kind of product not just websites and applications unlike UI that is only applicable in that context. It's true that UX is used heavily in designing web related applications but it also can be used to help design products from all kind of industries as long as those products are aimed at a target audience then UX can be applicable in helping to make a product more useful and enjoyable.

With that said, even though the two fields are different, both of them complement each other to provide an experience to the user that is meaningful in its context. A great UX design still would be bad if the visual aspect was poor. And a great UI design wouldn't make a user happy if the page was confusing and not clear of how the flow of user journey should be.

3.3. Bad design use cases

The importance of UX can be shown by looking at the biggest companies in the world who have chosen to invest time and money in having a dedicated UX team working on research and design to help making better products for their customers. Companies like Apple, Google, Amazon, and many more have made that choice and it clearly pays off. According to IBM the ROI (return of investment) on UX is astonishingly high where every 1\$ they have spent on UX have 10\$ to 100\$ ROI (Planet, How To Calculate the ROI Of Your UX Activities, 2017) (DESIGN, 2014). Therefore, we can clearly note that UX is not just a fancy trend or a luxury that only tech giants can afford to invest in rather it's a must have in any business that is involved in creating products for users.

But on the other hand, the importance of UX expands beyond money and ROI, bad design of products can actually be dangerous and have highly negative impact on people's lives. It expands beyond making a user frustrated or angry and not willing to use a product again, to actually harming people and in some cases bad design can lead to people's death.

In this section of the chapter we will take a look at few use cases from the book by Jonathan Shariat and Cynthia Savard "Tragic Design" (Saucier & Shariat, 2017) where bad design has led to dangerous situations where people's lives have been impacted that could have been avoided if there was good UX practice when designing those products.

Use case 1: Bad design that killed a cancer patient

A cancer patient in the united states after relapsing have entered the hospital and this time the medication prescribed for her was very powerful on the human body that it required 3 days of hydration process before the treatment and 3 days of hydration after the treatment. The nursing staff would input her information and status of her treatment in a charting system in the hospital to monitor her case and take actions accordingly, but due to the horrible design of

the software (see Figure 3) the nurses have failed to notice the 3 days of hydration after the treatment and the patient have died due to dehydration. The software lacks the basics of usability with crowded information everywhere with the important information that can literally kill someone is not highlighted or designed in a way that would be noticeable to the nurses or any user of the system.

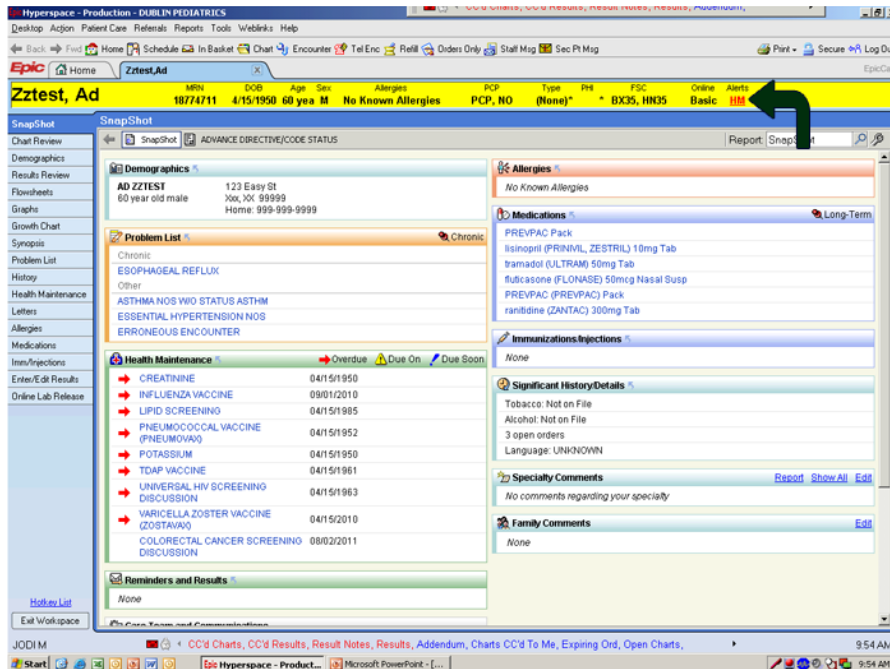


Figure 3: Charting system with bad design

The software covers all functionalities required that a user might need, but where it fails is taking into account the context of the users. As shown in an earlier section of this chapter (principles of UX) the design fails to understand that the context of the users here which are the nurses are very busy with many tasks outside the software, and that they need to do the charting in a timely manner. And having such design with a flood of information in one page with high complexity tasks for the users to do, the software design clearly fails to provide an easy experience to the specific user group it has.

A life of a human being has been lost due to a poorly designed application. If there was a UX team in charge of designing this software running usability checks and conducting feedback sessions with the users (the nurses) which would lead to iterative changes of the design, the mistake could have been easily avoided and the patient could be alive today.

Use case 2: Therac-25 Radiotherapy machine

This use case is from the 80s where UX design was not yet evolved as much as it is today, but the fatality of the design error is worth mentioning and learn from as the same UX principles that were not taken into account in the design of this machine could have been the reason to avoid the design error that have led to a patient death and other patients severely damaged.

This machine that was used in radiotherapy would blast the patient with a dosage of radiation set by the technician operating the machine but due to a design issue in some cases the machine would blast the patient with unprecedented high amount of radiation that could kill a person. Blasting a patient with 17000 rad (radiation absorbed dose) instead of 200 rad which would leave the patient in agony with the worst still to come over the next few weeks following the incident. The patient over the next few weeks of getting exposed to this huge

amount of radiation had paralyzed one of his arms, both his legs, and his vocal cords, and then died 5 months later. A tragic end that could have been avoided had there been a better design.

The issue that caused the machine to blast the patient with the wrong amount of radiation is that when the technician is manually entering the values in the interface (see Figure 4) of the machine have noticed that he have made a mistake and forgot to input the mode of therapy 4 lines above where he was currently in the interface and pressed the up arrow 4 times not knowing that the interface takes the up arrow for its value as a character and doesn't actually go up a row (the software only accepts keyboard entry and can't be operated with a mouse).

PATIENT NAME: John		BEAM TYPE: E		ENERGY (KeV): 10	
TREATMENT MODE: FIX					
	UNIT RATE/MINUTE	ACTUAL		PREScribed	
	0.000000	0.000000		0.000000	
	MONITOR UNITS	200.000000		200.000000	
	TIME (MIN)	0.270000		0.270000	
GANTRY ROTATION (DEG)		0.000000		0.000000	VERIFIED
COLLIMATOR ROTATION (DEG)		359.200000		359.200000	VERIFIED
COLLIMATOR X (CM)		14.200000		14.200000	VERIFIED
COLLIMATOR Y (CM)		27.200000		27.200000	VERIFIED
WEDGE NUMBER		1.000000		1.000000	VERIFIED
ACCESSORY NUMBER		0.000000		0.000000	VERIFIED
DATE: 2012-04-16	SYSTEM: BEAM READY	OP.MODE: TREAT		AUTO	
TIME: 11:48:58	TREAT: TREAT PAUSE	X-RAY		173777	
OPR ID: 033-ufs3p	REASON: OPERATOR	COMMAND: █			

Figure 4 The Interface of Therac-2

The main problems with the design here are, first it is clear that the up arrow cannot be an input that the user might want to input, and the interface have failed to inform or question the user about his mysterious input. Such simple message as “unexpected input” would have been sufficient to alert the user about his mistake.

The second interface design problem is that the field the user wanted to go up to and fill was still empty and the software in this case by design assumes a default value, which is a ridiculous design mistake taking into account that the radiation dosages are prescribed by the doctor are patient specific therefore the software allows default value which is not the prescribed value to this specific patient.

The third interface design problem is that after the user have thought that he finished filling the form and executed it an error message have been shown to the user “Malfunction 54”, which is not descriptive and does not explain what is exactly wrong to which the user have overridden the error as it was common to get all sort of similar errors that the users have grown custom to ignore for their reoccurrence and usually they are not fatal errors that would affect the workflow of the machine.

These errors could have been avoided if the interface was designed in a way that would provide the users with enough information about what they are doing wrong while they were doing it, if the designers took into consideration the reasonability of their choice of default values, and if the interface design would provide the user with a descriptive error messages that would explain what exactly is the issue to help the user understand the real problem.

Use case 3: A cruel Facebook feature design

In this use case the design doesn't cause the death of someone nor it causes a physical damage to a person, rather it caused an emotional damage and simply ruined someone's day and made them very sad.

A Facebook feature that was deployed by the end of 2014, a feature that is called "year in review" where it basically shows the users a summary of their year on Facebook including pictures and memories they had throughout the year. The feature lacked a very important principle of UX design which is addition of options that can be useful to the user while consuming the new feature. The year in review feature has lacked one important option which is to close or hide this feature rather than just seeing it on the top of the feed every time the application was used. What the UX designers at Facebook have failed to take into consideration are all the users who had bad times in the year and don't need the reminder every time they want to use Facebook. The design forced all users to keep seeing the year they had played in front of their eyes without any consideration that not everyone could be interested.

A hands on example of that is the case of a man named Eric Meyer who have shared his story where his daughter has passed away in 2014 and come the end of the year when he launched the Facebook application only to be reminded of his daughter passing away(see Figure 5), without the option of hiding this feature or closing it. This user ended up being reminded of his deceased daughter every single time he launched the app at the end of the year.

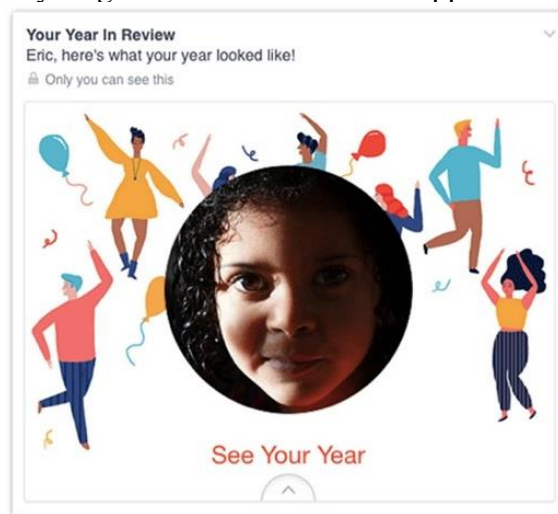


Figure 5: Year in review shared by Eric Meyer

Mainly two things the UX team at Facebook have failed to implement in their design in this case:

- 1- An option to close or hide this feature for those who don't need a reminder of their tough year.
- 2- To make it part of the design to take the context of the posts that are being added to the video, such as the tone of the comments, and the collection of the type of emotion in the post though the other people reacting to the post with sad reactions.

as shown in the principles of UX the last principle was for the UX designer to remember that there are humans in front of the interface. A good UX takes that into consideration and in this case the Facebook UX team have failed their users when they neglected to remember that some of their users' need and right is not to be reminded with sad events and even if they did,

to give them the option to hide it so it wouldn't cause a continuous emotional damage while using one of the most used applications in the world today.

3.4. Design thinking process

Phases of design thinking process

This section is based on chapter one of the book Design Thinking for Entrepreneurs and Small Businesses (Ingle, 2013)

The process of design thinking consists of 5 stages that are sequential and iterative as the following:

- Phase one: Understand
- Phase two: Define
- Phase three: Ideate
- Phase four: Prototype
- Phase five: Test

Phase one: Understand

In this phase the goal is to get a full understanding of the challenges that are facing the business in order to be able to create a solution for said challenges. Without understanding the real challenges or identifying the wrong ones could lead to a solution that ends up not addressing the challenge at hand. For example a business that is losing a segment of their customers after a competitor started offering a similar product with a lower price could think that the problem here is only the price and decide to lower theirs and ending up not gaining any customers back due to lack of understanding of what is the real challenge that they are facing and only looking at the surface of the issue rather than diving deep into determining what is their real challenge as a business. and in order to do so, an extensive research is needed. Without research it is hard to acquire data, and data is essential to understand the right problems and challenges that a business is facing. The research should try to cover all necessary aspects like customers and their needs, and competitors doing things differently and if it is successful or not. For the sake of our example here a research of the competitor could show that customers are switching to the competition because a better interface, or a more reliable customer support, and the price is not the actual challenge that was assumed in the beginning. Therefore, in the first phase in design thinking it is a must to leave all assumptions at the doorstep and dive deep into researching and collecting as much data as possible.

Phase two: Define

The "Define" phase starts after the conclusion of the "understand" phase after collecting enough data to begin the assessment of collected data to actually be able to define the root of the problem or challenge that the business is facing. Defining the right challenge is the key to solving it through the next phases of the process. It is important to note that if in this phase it was still not clear what is the main challenge and it was hard to point out and define clearly, that would be the result of bad research in the first phase hence this process is iterative and going back to the first phase is a must. This phase is known for one moment in particular which is usually referred to as the "a ha!" moment, the moment that the design thinker reaches a clear definition of what is the exact challenge that the business is facing

Phase three: Ideate

After reaching the "a ha!" moment, pinpointing and defining the root of the problem and the main challenge in the previous phase, the "Ideate" phase begins. In this phase the main task is

to come up with ideas on how to solve the problem. In order to do so the design thinker must conduct a brainstorming session or multiple ones depending on the need.

First the team of brainstormers should be identified. The team should contain individuals who correlate to the specific case at hand but at the same time it helps to have a diverse team from different ranks of the organization. In some cases, even people from outside the organization can be included such as vendors, customers, partners, or even family members but that should be decided by the design thinker who is tasked with creating the ideation or brainstorming team.

Once the team is formed and before the brainstorming session begins, the design thinker should remember:

- Disregarding typical business parameters such as feasibility, cost, resources, and so on.
- Ego should be left outside the session.
- There is no such thing as a bad idea.

Now when the session is starting the design thinker should communicate the following points to all participants in order to keep the session in track and not lose sight of what is the main goal of ideation session:

- Declaring the intention: a clear intention of the session communicated to everyone preferably beforehand would allow participant to be focused and prepared knowing exactly what the expectations of the session is.
- Every participant is equal: there could be in the same session a senior director and an associate but in this session both are equal. There is no higher authority to anyone over anyone else, and the only person who has authority in the room is the moderator of the session. Everyone is free to say what they want, and everyone is expected to listen to others speak regardless of the company hierarchy.
- Selection of the session moderator: the moderator's role here is to facilitate the session. Directing the conversation and making sure that it doesn't go off the rails, documenting everything that is going on in the session, and moving forward with the conversation if it gets stuck at one point for too long. The moderator of the session is preferred to have no interest of the outcome of the session and even preferably someone from outside the company for the session to be as unbiased as possible.
- Write all ideas down: documenting every idea that comes out during the session is one of the main goals of having such session. It cannot be left to chance and hope of remembering what happened or what was the outcome. A common way of doing so is by using sticky notes during the sessions and put every idea on the board and creating 3 categories with 3 different colors of sticky notes (ideas, thoughts, questions) and at the end of the session document the board by taking several clear photos of it so it's possible to come back to it after to get the results of the session properly documented.
- Silly ideas and comments: it's common that in the beginning of a brainstorming session for people to be nervous. Therefore, some of them might say silly ideas usually intended as jokes to break the ice and feel more comfortable. In case of that happening the design thinker and the moderator should let it happen as it is a normal behavior but, silly ideas should not be mistaken as bad ideas. As it was established earlier there is no bad ideas and the silly ideas meant here are just humorous attempts

by participants to feel more relaxed before getting serious. In case the silly ideas came back during the session it is usually an indicator that the session has gone too long and the moderator should start the process of ending it.

Phase four: Prototype

After the conclusion of the ideate phase there should be a good amount of ideas on how to solve the problem at hand and thus begins the prototype phase.

The first thing to do in the prototype phase is to bring back all the business parameters that were disregarded in the previous step to start evaluating what is possible and what is not in order to limit the number of ideas at hand. Once the business parameters are back in the picture the ideas can be looked at from 3 perspectives

- 1- Doable ideas
- 2- Amount of resources required
- 3- Potential impact

And then starting to put the ideas in these three categories and finding the sweet spot of ideas that is part of these three perspectives (see Figure 6). The number of ideas that satisfies all perspectives should be ideally more than 1 idea and less than 4. The rest of the ideas will be disregarded but documented for a possibility of future use and the small number of ideas that fit the caliber are prototyped with as much details as possible to prepare for the next phase which is the testing phase.

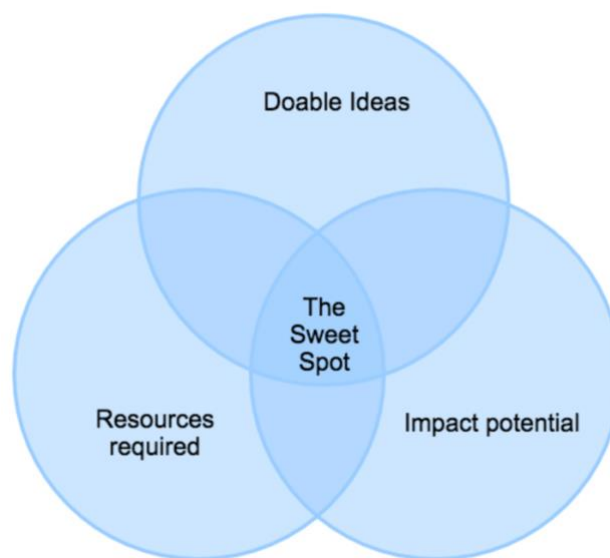


Figure 6: The sweet spot for ideas to prototype.

In the case of no ideas landing in the sweet spot described in Figure 6 above then the process goes back a phase to phase three as the process of design thinking is iterative and this activity is repeated with adjusted parameters until a better outcome is reached.

Phase five: Test

After completing a number of prototypes, it's time to put them into test to get to the one solution to the main problem the company is facing.

In the test phase, the first task to do is to identify the individuals that will test the prototypes to get the required feedback on what works and what doesn't. the individuals selected can be from different categories of people such as customers, technical employees, all kind of stakeholders. But, unlike in ideation phase, this time it's a must to separate the categories of

testers and get each of their feedback separately in order to get the perspective of each group individually therefore knowing what the customers think, and what other stakeholders think of the prototype. Getting the right people to test the prototypes is very crucial at this point and therefore there should not be any hesitation to invest in getting the right people to do so. Also, the design thinking team should provide a hospitable environment for the testers to get the best feedback possible.

Documenting every aspect of the testing sessions is a must to get enough data to get insights on each prototype. After collecting the feedback from each different category of testers, it's important to note the common comments from different tester groups as those comments are the high priority to take in consideration.

Then finally analyzing the results of each prototype testing sessions from each individual group to see which one performed the best with the focus on the main group that the solution should be created to address their need. Evaluating each prototype to the available resources in the company is also another factor to take in consideration in choosing the one prototype to be the solution chosen.

With the informed decision made based on real data, chances now are much higher that the solution have been found a viable business solution to address the company's challenge.

In case of a negative feedback on all of the prototypes then the design thinking team should take a step back and try to identify why that have happened in order to identify in which phase the error occurred and which phase they need to go back to and iterate to get a better results.

3.5. UX and design thinking models

Design thinking and UX are two fields that complement each other, and more often than not they are combined and looked at as one, Which is not a wrong way of looking at them where design thinking or UX research is responsible of understanding the challenges the business is facing then identifying the real problem then help find out solutions for it. UX design then takes the torch and helps with designing the prototypes and testing them then finally design the solution that will solve the problem and terminate or relieve the challenge.

The process of how this works was described in the previous section of the chapter (section 3.4) throughout 5 phases beginning with understanding and finishing with testing (see Figure 7).

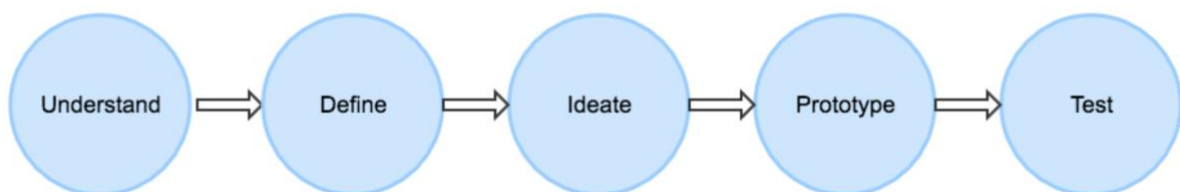


Figure 7: Design Thinking Process

In a nutshell, this process describes the phases that UX team (research and design) would follow to achieve their goal of solving business challenges or creating a product for internal use or for external customers. But doing a simple research would show that this process is not

followed as it is, and different businesses use different iteration of the process depending on what they see fit their business. These different iterations are simply different models of the main process in the figure above.

In this section we will look at different models and understand how each one works and how it differs itself from other models, but it's important to note that all these models somewhat follow the main process.

Zurb design thinking model

Zurb is a product design company that have started in 1998 in the United States of America with the focus on all digital products like web and mobile applications, websites, portals and desktop software. (Zurb, 2019) The company's customer's list include big names such Google, hp, ebay, and others (Zurb, Zurb Verify App, n.d.)

The company have their own model for design thinking and they call it Zurb design thinking model. the process of their model has been updated to a newer version for more clarity on how it works and a smoother transition from one phase to another. (see Figure 8) (see Figure 9)

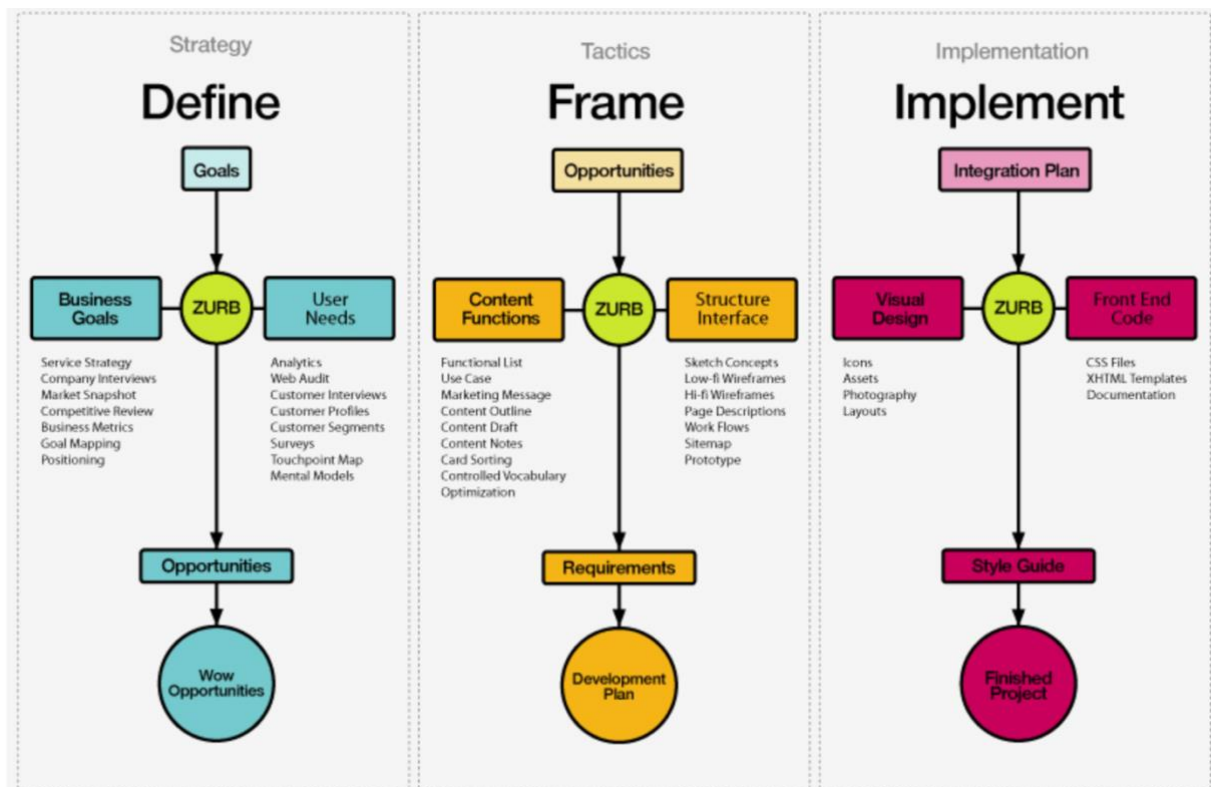


Figure 8: Zurb old design thinking model (Zurb, Design Process, n.d.)

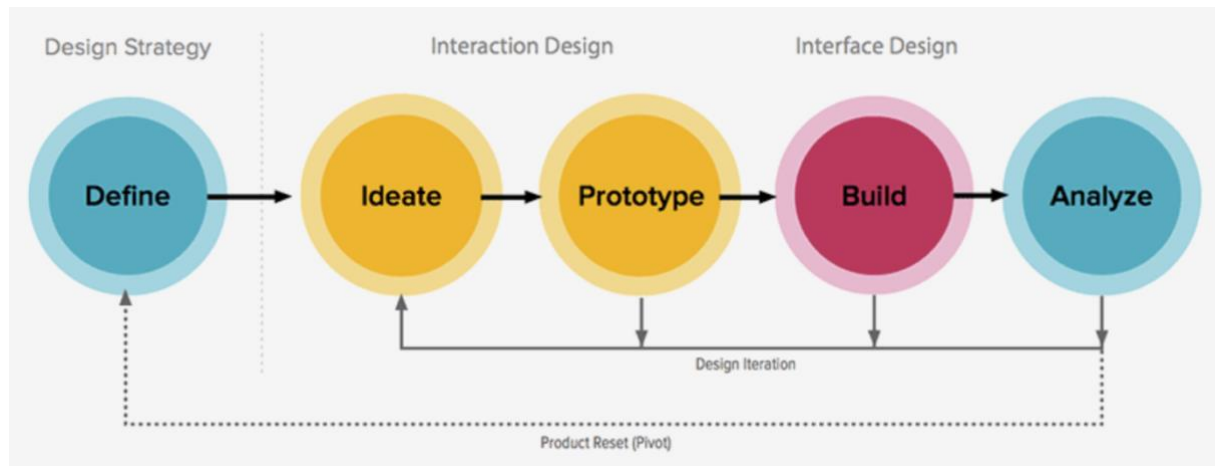


Figure 9: Zurb new design thinking model (Zurb, Dsign Process, n.d.)

Just from looking at Figure 9 we can see the similarities with the main process described in section 3.4 of this chapter but we can also see some differences. (Zurb, Dsign Process, n.d.) describe each phase as the following:

Define

in this phase the goal is define what needs to be built, finding user needs, and the goals that the customer would like to accomplish and the competitive purpose and positioning among competitors.

Ideate

In this phase Zurb's team works on creation of concepts, they do so by holding sessions where they sketch and draw ideas and developing the visual flow of the product. Those sketches should include content outlines and marketing messages.

Prototype

With the start of the prototyping phase, starts the validation of the result of the ideate phase and rapidly creating prototypes and test in on users to see what works and what should be left out. Zurb (Zurb, Dsign Process, n.d.) also recommends to try to do the effort and spend good amount of time on the frontend of prototypes to guarantee a better testing results.

Build

In this phase all the feedback and data analyzed and collected from the prototype phase are taken in consideration and the product development starts to build on the interface that was made during the prototype phase. By the end of this phase the product should be ready to be released.

Analyze

Finally after the product is released begins the last phase which is immediately after the release to start analyzing the end product from the perspective of how the users perceived it and collect as much feedback as possible to get insights if the product is a success and is what users and business needs. Iteration in the design phase and product tweaks are of a high importance in case of negative feedback.

That is how Zurb (Zurb, Dsign Process, n.d.) describe their model of the design thinking process and how each phase works.

This process in my opinion can work with only experts and experienced UX teams as its clear that it misses few steps on the way.

Negative thoughts on the process

First, jumping directly into defining the needs of the user and what the customer wants to build without extensive research beforehand on the root of the problem and what is the challenge that the product should address is fatal mistake as in most of the cases when customers get in touch with product design companies they come with an idea of what the end product will be which design companies should completely ignore and not have any bias on what the final outcome should be before doing their homework on identifying the right problem and then start to explore best possible solutions for it. The second problem with this model is that the process in general seems very product focused not problem solving focused which is in the root of design thinking. It is understandable that at the end of the day Zurb is a commercial company focused on getting clients and doing business but as a model of design thinking, it lacks some key aspects of the design thinking field and it needs more focus on challenges that needs solutions and humans rather than having a focus just on the product and productization

Positive thoughts on the process

The addition of the analysis phase at the end of the process is a positive way of evaluating the work done and it helps rapidly and quickly solving any issues in the product.

Double diamond model

The double diamond model is a design thinking model that was founded in the year 2005 by the design council (Council, n.d.) which is a nonprofit organization that was founded 75 years ago by the British wartime government to support Britain's economic recovery and until today their mission is to make life better by design. (Weir, 2019)

From its name the double diamond model is composed by two diamonds put together with the first diamond goal is to design the right thing, and the second diamond is to design things right. (see Figure 10)

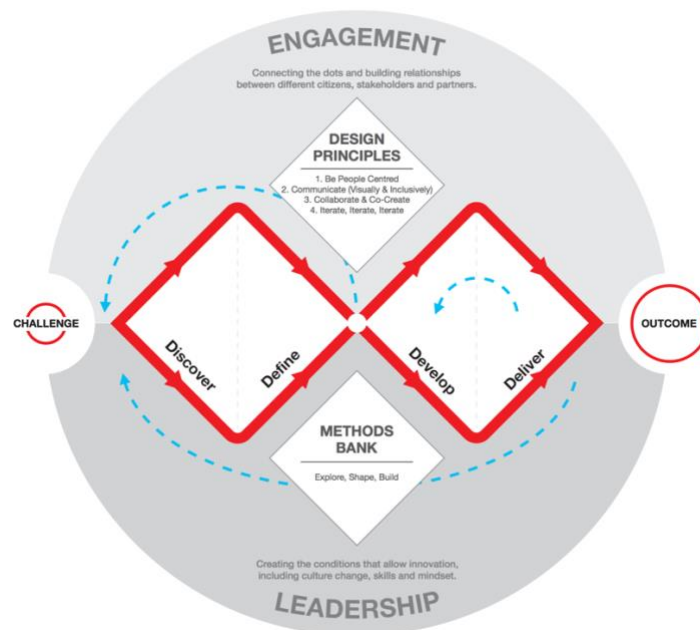


Figure 10: Double Diamond Model by design council (DesignCouncil, 2019)

(Council, n.d.) design council explain each phase of their double diamond model as the following:

Discover

This phase begins the first diamond and it is all about understanding people and conducting their research, leaving any assumption of the problem at the door and simply talk to the people who are affected by the challenges and issues to understand what the real problem is.

Define

In this phase the analysis of all data collected from the previous phase provides insights that help define the problem or challenge clearly.

Develop

The start of the second diamond is the develop phase after understanding the real problem from the first diamond. In this phase people involved are encouraged to provide their answers on how to solve the problem and find solutions.

Deliver

In this phase the solutions that resulted from the develop phase are tested and the ones that don't work are disregarded and the ones that work will be focused on and improved upon to finalize a solution that addresses the problem and fixes it.

The model in general is iterative and as seen in the figure above (Figure 10) the principals align with design thinking principles discussed in the previous section (section 3.4). The methods bank according to design council (Council, n.d.) are:

- Explore: challenges, needs and opportunities
- Shape: prototypes, insights and visions
- Build: ideas, plans and expertise

Thoughts on the double diamond model

This model in my opinion covers all the aspects of the design thinking process in a straightforward way, its simplicity is the strength of it. And most importantly the focus on people rather than products is the key for identifying the right challenge and problem to solve in an agile iterative way. The design council have really hit the nail with their design of this model as more often than not simplicity wins and works better than complicated designs and solutions, by looking at the model figure all its aspects can be understood clearly with little explanation needed.

The only negative thought I think of this process is the lack of emphasizing on the roles of develop and deliver as it's not very clear if the deliver phase is about delivering functioning prototypes/proof of concepts or delivering the final solution.

IBM design thinking model

By studying different design thinking models from different organizations, a noticeable pattern is identified with three basic concepts. The first one is discovery where the research is done, and all information is gathered to get a clear understanding and a problem or a challenge is discovered and identified. The second concept is to find out possible solutions based the facts and information from discovery. And the third is to make the solution and solve the problem.

The IBM model is no different in concept from the other model, and by now we can notice that any other model won't be different in concept but the naming of phases or number of phases might be different from one model to another depending on who created it as those creators made their models based on the experience they had in their work environment and iterations of past experiences.

As seen on the IBM design thinking web page (IBM, 2019), the IBM design thinking model consists of 3 phases, observe, reflect, make. (see Figure 11) It works as a loop with freedom to iterate and go back to any of the three phases as much as needed keeping the conversation always active with users and clients responding to every changing need through prototyping.

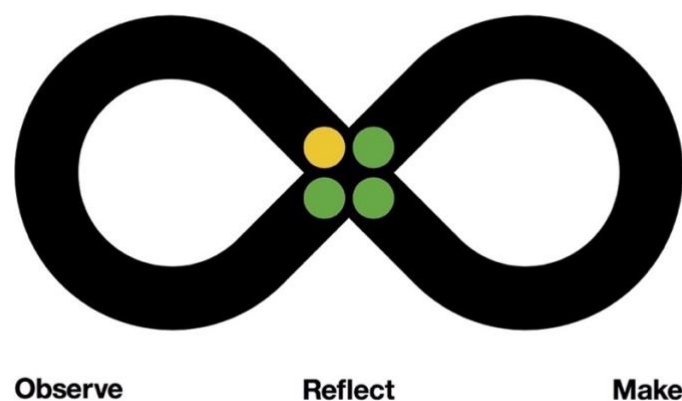


Figure 11: IBM design thinking model (model, 2016)

Observe

The loop starts with gaining a deep understanding of the challenges by a thorough user research to get to the main problem and to understand fully the user needs.

Reflect

Basically, in this phase the definition of the right challenge is cleared and the real problem to solve is chosen based on all the information and data gathered and analyzed in the observe phase. And then ideating possible solutions with brainstorming activities.

Make

An outcome-based phase with rapid prototyping of ideas and testing quickly and cheaply until reaching the right solution that is effective and well tested.

The IBM model is a loop that allows the rapid iteration and going back and forth between phases as much as needed to always strive for better results and to ensure that the right solution has been identified to the right problem.

Other design thinking models

There are many other models for design thinking out there, each developed by a different company or organization. These models may differ in names or in phases names and how they are visually represented but the one important thing in common among all these models is that they all are based on the same idea. Understanding and the challenge that the business or the customers are facing, defining the one true root of the problem, get as many ideas as possible

to solve the problem, design prototypes for the most promising ideas and test them, then finally choose the best solution in terms of problem solving, resources, and business needs.

Other models from companies like google or IDEO (consulting company) look different and feel different but it's all within the same design thinking process. Other models include: (Hoffman, 2016)

- Stanford design school model
- IDEO design thinking model
- Google design sprint model
- Austin center for design model
- SAP user centered design model
- Design for America company design model

Looking at all kind of different models (see Figure 12) to apply the design thinking process, I can say that using any of them would help solving problems in a systematic human centered way, but which one to use is up for debate depending on the context of the situation, people, expertise, and type of product.

The model I believe to be the simplest and makes sense to apply for those who don't have years of experience in design thinking such as myself is the double diamond model.

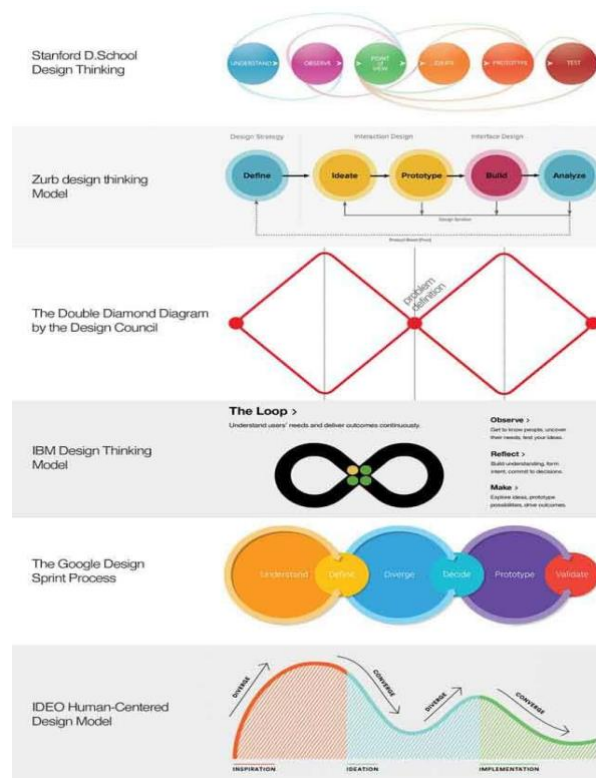


Figure 12 Other design thinking models (Scott, 2019)

3.6.Double Diamond Model

The double diamond model as the name suggests consists of two diamonds. The goal of the first diamond is to “design the right thing” which basically means to know what is it that we want our design to solve. And as I have mentioned more than once the cruciality of having the knowledge and figuring out the real problem or problems to the end solution will be what solves it. If the challenge choice was wrong the solution regardless of its design or the technology, it was built on won’t make people happy and no one out of your segment of customer or users would want it as it doesn’t add value to their lives by solving their problem.

The second diamond in the model is to “design things right” and that is basically finding possible solution with fast iterations of prototyping and testing until getting to the right one to implement and then reach the end with the development of a solution that will eliminate the challenge and solve the problem.

The output of the double diamond is not necessarily to be a complete product, it all depends on the scope, resources, and time set for the workshop where the model will be applied. The end result can be just an answer sketched on a white board if that was the scope set for the workshop and that was the expected delivery out of it, it can also be a complete product but in that case the length of the second diamond would be much longer with way more iteration and collaboration between the designers and developers and the business with multiple iteration to make sure that the end product is what the business needs and what the customers need.

It is also important to note that the double diamond is not a black box and it is perceived differently by different companies and different designers. It is dependent on the type of end result required and the perception of the model at hand at the moment is the one suited for designing products. For instance, if a workshop for process designing was held to create or modify a process based on an opportunity aka challenge the end result will not be more than few sketches that describe how the new process is going to work and the workshop wouldn’t be more than few days therefore the terms solution and prototypes when it comes to double diamond isn’t necessarily always a product.

In this section we will take a look at the double diamond design thinking process and explain how it works. Author John Whalen have explained it in chapter 17 “Succeed Fast, Succeed Often” of his book “Design for How People Think” (Whalen, 2019) and this section is based on the understanding of his explanation.

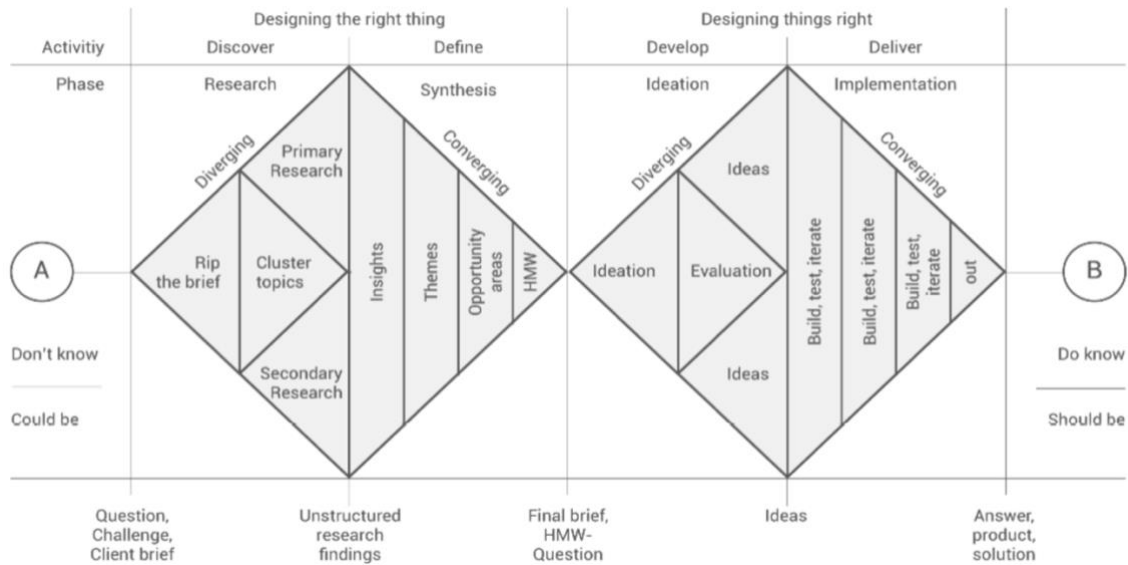


Figure 13 Double Diamond Model Phases (Whalen, 2019)

Starting point

In the beginning and before getting into the first diamond components there must be a customer that this model will work to satisfy. That customer can be an internal business department that needs help with a new or old product, it can be the management of company that needs help with finding what are the main challenges they are facing and how can they solve them, or it could be users that you are working toward creating a solution for to provide them a better experience in the context of their role in this equation. The important thing I would note here is not to listen to predefined problems before getting started so it wouldn't give you a biased perspective.

Discover

Starting the first diamond with the discovery activity, or as shown in Figure 13 the divergent part of the diamond. The discovery as an activity is about conducting a research to understand everything there is to understand. Conducting a research to find your customer needs and pain point and to empathize with them knowing what their pain points are and collecting as much data as possible. In order to do the research phase can be by talking directly to the customer with interviews, surveys, workshops, or any other means possible. The conducted research would help in identifying the opportunity areas hence the diverging, and then prepare for the second part of the first diamond, the converging.

Define

Now that the opportunities or problems have been identified the second activity in the first diamond is to define the opportunity or problem that will be the focus point of the next diamond. Defining the real opportunity occurs from insights found in the results of the research done the discovery phase. The most critical opportunities to the customer are selected. Note that the selection should be based on the research on the customer needs not what the customer might say. Once the main problem or problems are identified and funneled down the work of the first diamond done for now. Iteration of the process might bring the workshop back to the first diamond for better results incase needed.

Develop

Now that the problem is identified it is time for ideas on how to solve it. Ideation session with stakeholders including end-users, subject matter experts, and any other group of people that might have a different relevant perspective on how to solve the problem at hand. After the ideation is complete and there are plenty of ideas to go with, the ideas are put to evaluation from the perspective of resources, feasibility, and potential impact. The ideas that fit the caliber are the one that move in the process to the next activity which is delivery.

Delivery

Delivery can differ depending on what is the end goal of the model. the main idea here is implementation of ideas, but the degree of implementation differs depending on the context of the design thinking workshop. The delivery in a more general way should be a complete product or solution therefore there will be many iterations and prototypes and collaboration between UX designers and developers alongside the business to keep in check that the business or customer requirements haven't changed. Starting with prototyping the 2 or 3 ideas that fit our ideas and then from those prototypes create a proof of concept and keep iterating the diamond process (building, testing, iterating) until reaching the final solution or the final product thus getting to the finishing line.

Other interpretations of the double diamond model would reach the finish lines with the end results being just concepts of what the final solution should look like. It all depends on what the expectation is from the customer.

The author (Whalen, 2019) have a theory by the name of the six minds framework to help UX designers and design thinkers narrow down the number of possibilities and the number of ideas to implement through understanding six functionalities of the human mind:

- 1- Vision/attention: answering the questions about what the audience is looking for? What attracts their attention? And what are their expectations of pictures and words? Helps the designer to use this information to their advantage and taking them into account when designing what the user should or shouldn't see.
- 2- Wayfinding: understanding what the expectations of the user about the journey in the product helps designing a product the users can interact with and find their way through it easily without frustrations.
- 3- Memory: designing a product that the user interacted with a similar product design in the past helps to speed the product acceptance and build trust with the product.
- 4- Language: to speak the language of the user and know to which extent the user knowledge in an area to be able to design a product that the user understands.
- 5- Problem solving: clearing the process of what the user needs to do is key for them to be able to use the product to their expectations without being lost or unable to solve a problem.
- 6- Emotion: the product or solution should eliminate the user fears by achieving their short-term goals, in order to get a chance to have consistency to reach long-term goals.

4. The Data Science Portal

In this chapter we will take a look at the data science portal, **the case study for this paper**. The goal of this chapter is to understand the product and know what service it provides to its users, and what are the features of it before diving deep on how it was created and what role UX and design thinking played in the creation of it in the next chapters.

First of all, a small brief on the company. As mentioned on the company's website "MSD is a leading global bio-pharmaceutical company and delivers innovative health treatments for the world's most challenging diseases including cancer, Alzheimer's, cardio-metabolic, and infectious diseases including HIV and Ebola as well as animal diseases. Known as Merck & Co., Inc., Kenilworth, NJ, USA in the United States and Canada, we operate in 140 countries and have been in the Czech Republic since 1992". (MSDIT, 2019).

The Data Science Portal is an internal product in MSD that is a knowledge base intended for all data scientists in the company, providing a single source of knowledge around internal processes. The portal provides a wide variety of articles where each one answers a specific question related to a data scientist work in the company. The portal is also a place where data scientists can learn regardless of the scientist seniority or experience something new directly related to their work or processes and also learn about aspects of projects from A to Z ranging from projects demand management to projects operations and support.

Most importantly the Data Science Portal is a community for data scientist to learn, share, and improve their knowledge on anything data science related. It also provides the data scientists with the ability to add their own articles and share their knowledge with other scientists to maintain a healthy culture of knowledge sharing in the community, and in case a topic or question was not covered in the portal, data scientists have the ability to request an article to let the Data Science Portal team know the need of a new article for them to get started on creating it which the process of how that happens will be explained in a later chapter.

Data Science Portal categories

The data science consists of nine main categories that cover all important aspects of projects and the process of work for a data scientist. Each category contains subsections with explanation about the subsection and under each subsection the user can find articles that answers questions in that area, contacts, and useful links (internal and external).

The categories of the Data Science Portal Are:

- Demand management for IT resources: this category covers few areas of demand management related to data science and with resources and links related to demand management. The category is divided into two subsections, the first one is about client service leads, and the second one is about demand pipeline. With each section having articles, contacts, and links.
- Business problem identification: this category helps the user learn about UX discovery in the company and learn about find information about the company and the pharma industry, and the last subcategory is about data privacy and ethics.
- Setting up analytics environment: one of the major categories in the Data Science Portal since it is directly related to the data scientists work. This category has 4 subsections as the following:
 - R programing language

- Python programming language
- Visualization tools
- High performance computing
- Preprocessing and modeling: this category contains a section for technologies and tools, a section for scheduling and pipelines, and a section for modeling techniques.
- Collaboration and application development: in this category the subsections are:
 - Code versioning and sharing
 - Application Development
 - Automated build, testing and deployment
- Visualization and dashboards: the user can learn in this category about different tools, scripting, and connections related to visualization with a variety of articles and useful links.
- Industrialization/Productization: in this category the user can find articles in 3 different subsections about SDLC (System Development Life Cycle) documentation, Architecture of systems, and APIs (Application Programming Interface). (see figure 4-3)
- Operations and support: the last category in the portal the user can find articles to learn about application monitoring, user's behavior, and vendor assessment.

It is important to note that those categories covers all possible aspects of any project in the company giving data scientists to obtain knowledge not only related to their day to day work but also provides the possibility to obtain information that usually takes years of being in the same organization to know thus decreasing the time needed to onboard new data scientists and increase the knowledge of the ones who have been working in the company for a while.

Other categories or sections of the Data Science portal include the “**About us**” page that provide to the users more detailed explanation about the scope and vision behind the Data Science Portal, and also include the instructions on how to create an article and how to request an article, and also on the “About Us” page users can see the team behind this product and how to contact them.

The Article

Articles are the main product of the Data Science Portal. It is what gives the value to the user by learning or finding all kind of answers to questions that might rise during the course of the user's work. Those answers are in the form of articles and a step by step guides that the user can read and learn how to perform a task or how a specific process works inside the company.

All the articles to a possible degree follow the same format to make the user get the information he/she seeks faster once the user have seen few articles to get familiar with the format. Each article has the following sections:

- Problem Definition: indicates the area or the problem that the article will be talking about.
- Solution overview: the solution or the explanation just in few sentences, and it is intended for those who are familiar with the topic and need a quick reminder.
- Complete solution guide: in this section the user can find the complete solution or answer to the question that the article title asks solved in a step by step guide sparing no detail so the user can apply the same and solve his/her problem or learn how a process works in details.
- Additional information: a section for adding all internal and external related links to the article.

- Article owner: this section shows the user who is responsible for this article with all contact details in case the user would like to contact them and learn more or ask more questions related to the source of the information directly (the article owner).
- Related articles: showing the user the other articles related to the one he/she is reading.
- Comment section and feedback: users can like and comment on each article directly or they can click the feedback button and send feedback to the Data Science Portal team.

Note that screen shots of the Data Science Portal can be found in the appendix A of this paper.

5. Web Technology

In this chapter we will take a look at the Data Science Portal from a technological perspective and describe all the web technology that was used in building this product.

The Data Science Portal as an internal product was built on top of Confluence platform provided by Atlassian technologies (Atlassian, 2019) that provide multiple tools and platform to the company such as Confluence and Jira which are both used in building the Data Science Portal. Leveraging these tools for this product will be explained in detail in this chapter.

The Data Science Portal consumes the following technologies:

- 1- Confluence
- 2- Jira
- 3- HTML, CSS, and JavaScript

5.1. Atlassian technologies: Confluence

Confluence is team collaboration tool that is a wiki like platform that allows teams to collaborate on projects and keep track of their work, plan meetings, and document and share knowledge across internal teams. It is an internal platform in the company that is used due to its simplicity, speed and ability to connecting to other platforms such as Jira and Bitbucket.

Confluence main features (Confluence, 2019):

- Allows multiple authorized user editing at the same time and see changes in real time
- Provides users with the ability to write inline comments to give accelerated feedback
- Provide users with a variety different templates to creating pages quickly for daily documentation
- Support for Mobile phones and automatically syncs between mobile and desktop versions.
- The ability to create workspaces for each team, department, or project to help organize the flow of information sharing.
- Page and file versioning to keep track of changes and not losing any data
- Users can subscribe to a specific workspace or page to get email notifications about changes and updates.
- The ability to restrict a workspace or a page to a specific group of users
- Additional functionalities through plugins which are called macros in confluence to enrich the content created.
- Confluence is highly customizable through macros.
- The ability to customize workspaces and pages through web development and coding in different languages including JavaScript, Python, and many others using the macro **code block**
- The ability to integrate content from one workspace into another using the macros
 - Include page for full page integration
 - Excerpt
 - Excerpt include for a partial integration of content from one workspace to another.
- Allows users to integrate tasks assigned from Jira and track their status on confluence.

Workspace

A workspace is a directory of pages that allow specific group of users to create edit and organize different pages and share information about a team or a project in a meaningful way. Each space has a home page and a tree of subpages that users can add to and organize the tree of pages according to their need. (See Figure 14)



Figure 14: A simple Confluence workspace (Confluence, spacetree for confluence, n.d.)

For the Data Science Portal an empty workspace was created to build the product on top of it capitalizing on the variety of plugins aka “macros” available to enrich the content and the visual aspect of the product and also capitalizing on confluence’s customization endless possibilities through coding with HTML and JavaScript.

Confluence Macros

Macros are basically plugin in confluence that helps extending the capabilities of a confluence page and adds functionality to it by selecting a macro from variant categories while editing or creating a page like the HTML macro and the Code block macro that enables the users to code design and functionality into their pages and workspace, or macros that enrich the content and visual aspect of the page. (see Figure 15)

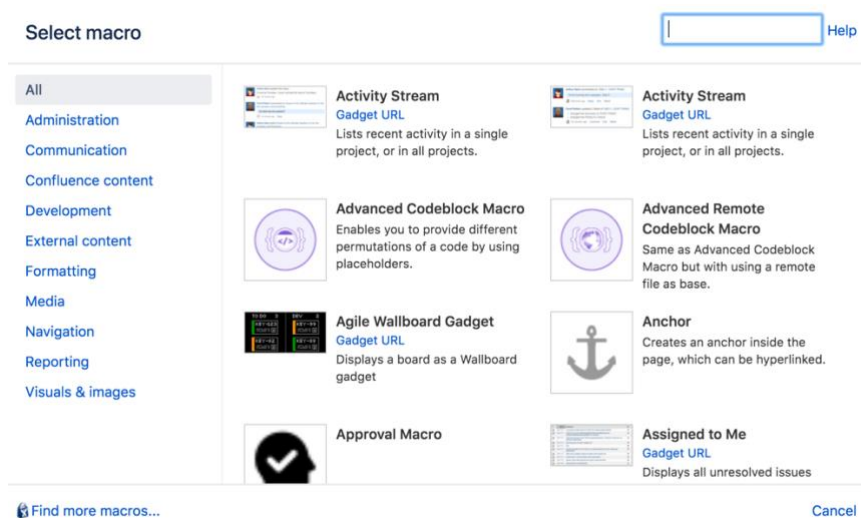


Figure 15 Macro selection menu on Confluence

Important macros used in the Data Science Portal

The out of the box wide range of different macros that are available in confluence have covered a number of functionalities in the Data Science Portal without the need for coding which helped reduce the time needed for the implementation of the project, and the only coding done was to make some tweaks for design purposes or add few functionalities that confluence macros do not offer.

The first crucial macro to the success of the project is a set of two macros that work together:

- Excerpt
- Excerpt include

Those two macros allow to take part of a page on a different workspace and add that specific part to another page in a different space which was an important aspect of the process to creating articles in the data science portal that will be explained later in chapter 7.

The idea behind using those two macros is to keep the main answers that the articles provide kept and edited by the subject matter experts in their own team spaces and the macro allows to take that part and include it in the article on the data science portal while maintaining the portal interface design specification with the excerpted and included part looking like part of the article page without any indication that it is part of another page in a different space. If the subject matter experts updated their processes or updated the excerpted part of the page on their team space that contains the answer in the article, then it would automatically be updated in the article in the Data Science Portal.

Other macros important macros that add functionality to the product are:

- Search macro: adds a search bar in the page with customizable settings to enable users to search content in the portal.
- HTML macro: allows adding a HTML or JavaScript to a page or workspace. Mostly was used in the UI design, calling the CSS file, and adding the JavaScript to add functionality to the feedback button.
- AUI Button macro: allows adding a button with either an interface to define what clicking the button does, or the ability to add the button to be part of a script where the functionality is coded.
- Content by label macro: this macro allows the listing of pages that have the same label and it was used in the articles to show related articles to the users when they are reading an article

Other macros where used for the arrangement of the pages but the ones listed above are the main confluence macros that adds functionality to product.

How to set up a local R environment

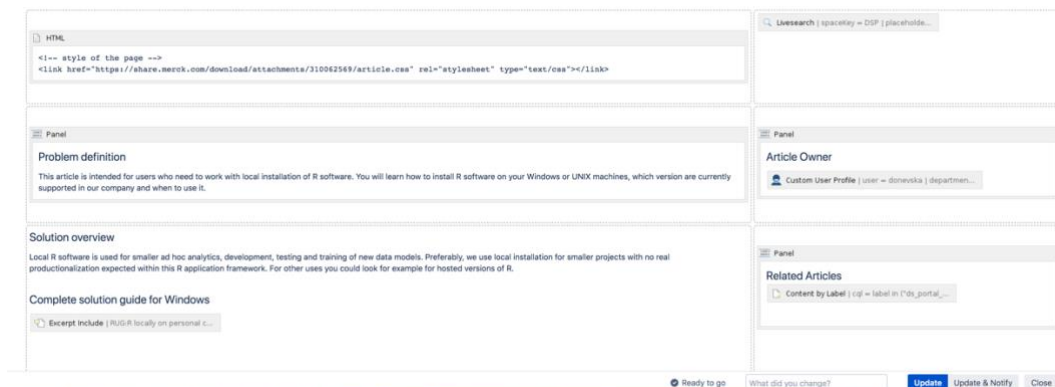


Figure 16: examples of macros used in an article

5.2. Atlassian technologies: Jira

Jira is part of the Atlassian family of products. It's an agile project management platform for tracking project issues and tasks (Jira, 2019). It offers two ways of agile project tracking through project boards.

Scrum boards for projects that need iterative and incremental delivery. The scrum framework enables software teams to manage complex projects and creating a culture of collaboration (Jira, scrum boards, 2019). With each iteration taking roughly between 1-3 weeks.

Kanban boards for projects that focusing on the big picture, providing a full visibility into the next step to achieve delivery in one output cycle.

Jira's role in the Data Science Portal

The Jira platform had two important and different roles:

- The first role is in the content creation tracking: tracking the articles that are being made. The portal started with a set of questions collected from the data scientist during the design thinking workshops that will be discussed in chapter 7, and each question was tracked as a ticket with different stages. Each ticket starts in pending column and once the process of creating the article that addresses this question the ticket moves to in progress column, then it is reviewed by the subject matter expert and moved to the in review column, then when the expert approves, the article is published on the Data Science Portal and the ticket is moved to the published column. The other way content is created and tracked is by the “**Request an article**” button on the portal where the user types a question and clicks send which automatically creates a ticket in the board with the question and Jira notifies the project administrators so they can start working on creating an article to address the requested one.
- The second role of Jira in the portal is collecting feedback from the portal visitors (users). The users have the option in all portal pages to click on the feedback button and when they do and click send a ticket is created in Jira in a secondary board with the ticket having information about the user and the page in order to know what action to take depending on the feedback in case it was a negative one.

Issue collector

The issue collector feature in Jira (Figure 17) can be done from the Jira project settings. It has a UI that lets the project administrator choose the fields and type of feedback he/she wants to collect with few feedback templates and there is also an option for customization where Jira would provide the necessary JavaScript for the connection to the project board and the administrator can implement it in their code. Once the issue collector is created Jira provides the code that can be then inserted in the HTML macro in confluence and the button for feedback will be created and connected with the Jira project to create tickets for every feedback given through the button.

Currently there are two different types of issue collectors and two different types of feedback on the Data Science Portal:

- General feedback on the home page that users can report what they like and what they don't like about the portal. (see Figure 18)

- Article feedback where users can give feedback on a every article. Users have to choose the problem type whether it is related to the article itself or it is about the process that the article describes. (see Figure 19)

Issue collector setup page on the Jira project:

Summary

Details

Subtask Templates

Issue types

Assignee

Bug

Change Request

Epic

General Request

New Feature

Requirement

Risk

Story

Sub-task

2 more issue types

Workflows

Screens

Fields

Priorities

Versions

Components

Users and roles

Permissions

Issue Security

Notifications

Project automation

Development tools

Issue collectors

Add issue collector

Name*

Description

Issue Type*

Story

Feedback will be created as issues of this issue type

Reporter*

Start typing to get a list of possible matches.

Reporter for the issue. Used unless a reporter is matched to a Jira user with create issue permissions.

Match reporter?

☒ Always use issue reporter
 ☐ Attempt to match user session of submitter or submitter email address

Collect browser info

☒ Collects the environment data of the user, if they consent to it being collected. This data includes the browser type, screen resolution, referral header, and URL where the feedback was collected.

Trigger


Trigger text*

Provide feedback

Trigger style

☒ Prominent
 ☐ Subtle
 ☐ Vertical
 ☐ Custom

Trigger preview



Issue collector form

Template

☒ Got feedback?
 ☐ Raise a bug
 ☐ Custom

Message

Please provide your feedback below:

Template preview

Got feedback?

Please provide your feedback below:

Rate this page*

☐ Awesome!
 ☐ Good
 ☐ Meh!
 ☐ Bad
 ☐ Horrible!

What do you like?*

What needs to be improved?*

Submit

Close

Figure 17: Creating an issue collector on Jira

41

General feedback:

Got feedback?

Please provide your feedback below:

Rate this page* ☐ Awesome! ☐ Good ☐ Meh! ☐ Bad ☐ Horrible!

What do you like?* this is useful

What needs to be improved?* need more content

Attach file Choose Files No file chosen

Name Ahmad Raslan

Email

Submit Close

Figure 18: General feedback prompt window Article feedback on the Data Science Portal and associated Jira ticket

Report an issue

Please provide your feedback below:
(Specify whether you are giving feedback regarding the article or regarding the process that the article describes or both by selecting from the "Feedback type" menu)

Feedback type Article

Summary Process

Details This process is outdated and needs to be updated

Your Name Ahmad Raslan

Your Email

Submit Close

Edit Issue : DSPO-98

Summary Outdated

Issue Type User Report

Component/s Process

Assignee Unassigned

Reporter Raslan, Ahmad

Description This process is outdated and needs to be updated
Reporter: Ahmad Raslan

Attachment Drop files to attach, or browse.

Linked issues blocks

Update Cancel

Figure 19: Article feedback prompt & associated Jira ticket

5.3. HTML, JavaScript, CSS

One of the main reasons to build the Data science Portal as a confluence workspace is that confluence's ability to understand code, and it is possible to customize the workspace in anyway needed by coding. The coding used in the data science portal was wrote in three parts, on each page as HTML and JavaScript to add specific functionalities and design aspects that a normal confluence workspace wouldn't usually offer, and the third part is the CSS file which confluence has as a feature to add a CSS file to the backend of the workspace to enhance the looks and feels and to have a specific design features on the workspace level regardless of which page is being viewed by the users.

In this section we will take a look at each element to explain what is each one briefly, and to show the code that was written to implement the Data Science Portal and explain what each code block functionality is, then explain the CSS file of the whole workspace.

HTML

According to w3schools one of the biggest sights for learning HTML, HTML is the abbreviation for Hyper Text Markup Language, it is basically the language of the web that describes the structure of the web page (W3schools, 2019).

HTML consists of elements that are called usually tags. A tag usually looks like this " <Name>", a tag usually has a start "<Name>" and it has an end "</Name>". The tags or the elements of HTML that basically tell the web browser how to display its content. The browsers render the HTML tags and display the content without displaying the tags themselves. (W3schools, 2019).

CSS

According to w3schools, CSS is the abbreviation for Cascading Style Sheet and it is the language that describes the style of and HTML elements and how the elements should be displayed (w3schools, 2019). The CSS file can save time when creating websites to have a general style layout such as colors font and etc..., the CSS file is saved in a location then called in the website to load the style to the website without the need to redo the styling every time.

JavaScript

JavaScript is one the main programming languages mostly used for web development. The reason for this is that JavaScript works perfectly with HTML and CSS to give functionality to the webpage. JavaScript code must be contained between two tags, <script> and </script>.

HTML, CSS, and JavaScript were used to fill the gap of design and functionality that confluence lacks as a platform. The main idea behind that was to make the user look at the portal and get the feeling of a complete product and not just another workspace on confluence but rather a place where the user have trust in its content and can rely on it with the added functionality such as requesting an article, giving feedback, and creating a new article to grow the portal even bigger. And now the code of each page will be displayed in the following pages.

The Data Science Portal Code (HTML, JavaScript, CSS)

The Data Science Portal Home Page:

This code snippet is for hiding the comment section from confluence

```
<script type="text/javascript">
AJS.toInit(function() {
AJS.$('#comments-section').hide();
});
</script>
```

This code snippet is for giving functionality for the AUI button which is a macro in confluence and its ID is “Feedback” so it shows the dialog box of the issue collector for the user to be able to give feedback

```
<script type="text/javascript">
    window.ATL_JQ_PAGE_PROPS = {
        "triggerFunction": function(showCollectorDialog) {
            jQuery("#FeedBack").click(function(e) {
                e.preventDefault();
                showCollectorDialog();
            });
        }
    };
</script>
```

The Data Science Portal “About Us” page:

This code snippet is for hiding the comment section on the page

```
<script type="text/javascript">
AJS.toInit(function() {
AJS.$('#comments-section').hide();
});
</script>
```

This code snippet is for giving functionality to the AUI button created by confluence macro and has the ID “RequestArticle” to open the issue collector dialog box

```
<script type="text/javascript">
    window.ATL_JQ_PAGE_PROPS = {
        "triggerFunction": function(showCollectorDialog) {
            jQuery("#RequestArticle").click(function(e) {
                e.preventDefault();
                showCollectorDialog();
            });
        }
    };
</script>
```

The Data Science Portal category pages:

All the categories have the same script mentioned above in the “About us” page to request an article and to call the CSS file and the rest of the page is done with confluence capabilities.

The Data Science Portal Article page:

This code snippet is for giving the functionality to the AUI button created with confluence macro with the ID “ArticleFeedBack”:

```
<script type="text/javascript">
    window.ATL_JQ_PAGE_PROPS = {
        "triggerFunction": function(showCollectorDialog) {
            jQuery("#ArticleFeedBack").click(function(e) {
                e.preventDefault();
                showCollectorDialog();
            });
        }
    };
};
```

This code snippet is for including the URL of the page when the user gives feedback by creating a custom field in the Jira issue collector and calling it here to include the URL whenever a user uses the article feedback button:

```
<script>
var articleURL = window.location.href;
window.ATL_JQ_PAGE_PROPS = $.extend(window.ATL_JQ_PAGE_PROPS, {
    // ==== we add the code below to set the field values ====
    fieldValues: {
        'customfield_13401' : articleURL
    }
});
</script>
```


6. Description of the problem and the business need

For any product that is ever created there must be two main elements, the first is that the product should solve or help solving a problem, and the second that there should be a need from the business. The business can be external customers, and it also can be colleagues or another team in company and in this case the colleagues become the business or the customers.

In this project the primary customer is the data scientists in MSD across different locations. The problem or the business need is simply to have more support for data scientists in their day to day job and as the simple main goal is to **make data scientists life easier in the company**. At the end of the day the idea was always to have a product that will contribute to all data scientists in all the three different locations but in an early stage it is hard to tell what this product should be? Should it be built internally, or should it be bought from a vendor?

To answer these questions, it is important first to focus on the main target who should benefit from any solution that will be implemented and the target is of course the data scientists in the company. And to do so a design thinking approach shall be taken to get to the right problem and make the right solution.

In this chapter and the following one we will see how the Double Diamond model of design thinking was applied to build the Data Science Portal as the right solution to support data scientists in the company across different locations to achieve the goal of making data scientists life easier in the company.

As mentioned in chapter three the double diamond is two connected diamonds where in the first one we diverge on problems (discover) then we converge on the right problem (define) based on the research done in discovery, then the second diamond we diverge on solutions (develop) and then converge(deliver) the right solution. This is the basic version and common version of the double diamond, the approach used in this project is based on this model with few adjustments to fit the use case at hand.

Discover

In this first phase the goal is to discover what are the problems and empathize with the pain points of the data scientists in the company. In order to achieve that, a session was held with data scientists across different locations of the company and the collection of general problems and issues data scientists feel they have in the company.

The discovery session focus was on two main points that data scientists were given the platform to share. The first point is the **Pain Points**, and the second is **Needs** of data scientists and data science teams in the company across different locations of the company.

As a result of this workshop a total of 31 **Need** and 23 **Pain Points** were collected that data scientists have shared during the workshop. **For confidentiality reasons the raw data of these sessions cannot be displayed or shared in this paper**. The results had a clear theme as most of what was shared can be categorized a pain point as “don’t know” or need as “need to know”

Define

After the discovery workshop and in the define phase the data collected from discovery is analysed and short listed to 17 areas that mostly revolve around lack of knowledge sharing. The list of problem areas cannot be displayed or shared in this paper for confidentiality reasons.

The outcome of this phase should be the right problem or problems to solve by creating a product or a prototype in later phases.

To identify the real or the right problem/s we go back to the results collected from the discovery session with data scientists and try to identify a connection between more than one problem in order to get a sense of the real challenge facing the data scientists and try to identify an opportunity to productize a solution to help solving the challenge.

The first insight from the data collected is the lack of information provided to data scientists related to the scope of their work and outside of it. We get to this conclusion by looking at some pain points or needs and dig deep and research them internally and find out that what was said to be not available or is said to be a need is actually available in the company but the information has not been communicated properly or the information exists somewhere in the company but data scientists do not know of its existence.

Another insight is the integration of newly hired data scientists into productivity. To integrate new hires to the work environment and getting them to know the ins and outs of the work environment is time consuming and something needs to be done to shorten that time period.

Some problems are also ignored as they are by design. For example, the lack of control over purchasing licenses by data scientists as buying a license or a service from a vendor should follow the process set by the company and not be available for anyone just to purchase without going through the right channels. What is in scope in this area is the information of how to follow the company process to purchase a software licenses or a service to be compliant and to have the job done fast without complexity.

From the insights found from the data collected the following problems are identified as the right problems to come up with ideas how to solve:

- Not enough visibility on what is currently available on the platforms for data scientists
- Sometimes it is hard for new hires to integrate - overwhelmed with the enterprise, platforms, complex infrastructure
- Hard to choose the tools data scientists want to use per solution

These three problems will be the input for the next phase in the double diamond model and the beginning of the second diamond.

Develop

Now it is time to ideate what can be the possible solution for the chosen problems that has been identified as the right problems to solve. In the ideate session with different stakeholders including data scientists, data science team leads, software engineering team leads, and data science management, there was a sort of agreement that the right solution would be a knowledge base that would:

- Provide information to data scientist in regards the available platforms in the company and how to utilize them in their work
- Provide new hires with all sort of “how to” content to help integrate them faster into their work
- Provide data scientists with tutorials to available tools in the company and when to use what

The solution itself is clear at this point which is to have a knowledgebase dedicated to all data scientists across the company but there are different options to consider:

- Get a knowledgebase vendor solution
- Build a knowledge base system internally from scratch as an internal website
- Capitalize on the company’s platforms like Confluence and SharePoint and build a knowledge base on top of one of them as they are the platforms for sharing content among employees in the company.

At the end of the develop phase of the double diamond the output is 3 different solutions. In order to decide which solution will be prototyped or implemented we need to go to the next phase.

Deliver

In this last phase of the double diamond the input is the three ideas developed in the last phase and it is necessary to figure out the feasibility of each one and those feasible ideas are then prototyped and tested on users and iterated based on the feedback provided by users.

The first idea was to get a knowledgebase from a vendor

The IT teams in the company are business supporting, therefore any project that is implemented should provide direct value to the business and in the case of the knowledgebase the value that it provides to the business is by supporting data scientists to be happier and do their jobs easier by which will improve their performance therefore providing value to the business. with that said this solution would take a considerable amount of time to find the right vendor. This solution idea would have high price and high time to market. Another important point is that the stakeholders have decided to leverage an internal solution there for this idea is rejected.

The second idea was to build an internal knowledgebase website

This is a feasible idea to implement and create a new product internally from scratch. But the problem with this idea is that it would take a large effort and would require a big team to implement in a short time, and again the stakeholders’ main objective to have as much of a fast solution as possible. Such idea would require a joined effort from different IT teams, although this is a feasible solution that would solve the problem, but it might not be the best one.

The third idea is to capitalize on the company’s platforms like Confluence and SharePoint and build a knowledge base on top of one of them

This idea as simple as it seems it is the most feasible one for many reasons as the following:

- Almost no resources cost on the data science team due to the fact that these two platforms are used at a companywide level and their cost are covered regardless of what is created on top of them
- No need for software or hardware maintenance from the data science team as the platforms are maintained by other IT teams

- Both platforms have the capability to cover the aspects of a knowledgebase
- Every employee in the company have access to both platforms by default therefore the accessibility will not be an issue
- It would be easy to implement in a short time

This the most feasible solution out of all three discussed and therefore is the one chosen to prototype. The first step would be to select the platform either SharePoint or Confluence in order to start building what would be the prototype or the minimal viable product (MVP).

Platform Choice

The chosen platform to build the knowledgebase on top is Confluence for a simple fact that does not require much of a research as it is common knowledge in the company that IT users primarily use Confluence and business users primarily use SharePoint. Therefore, since the users here are data scientists hence IT users and they do commonly use Confluence way more noticeably than SharePoint, the decision was to build this knowledge base on top of Confluence.

Another point in the favour of Confluence over Microsoft's SharePoint is the amount of customisation possible on each platform, Confluence definitely offers more customisation options therefore more development and design choices would be possible with Confluence for the purpose of the required solution.

First prototype

At the end of the deliver phase in the double diamond model the solution or product prototype should be delivered. The amount of development in the prototype differs from one project to another or even from one company to another. In this project's case the prototype is an initial sketch of how the knowledge base should look like in order to get feedback or approval from stakeholders in order to begin the project of building the knowledge base.

The first important UX principle that was taken into account when sketching the prototype is to **keep it simple** and straight forward. The knowledgebase will be divided into categories browsable by the user. The knowledge base should indicate to the user that this is a place to learn from to make the purpose of the product clear and to make sure the user understands what his role in his/her journey once he/she reaches the landing page. Another UX principle taken into account **Favour the familiar**, the design should not be out of the ordinary and the users should understand how to operate and get what they need quickly as it is a design that is very similar to most what users have seen in many internal and external websites.



Figure 20 Data Science Portal prototype sketch

7. Building the Data Science Portal

After the conclusion of the design thinking workshop applying the double diamond with the end result being the prototype sketch (see Figure 20), the double diamond is not fully done yet as it will be useful in actually during the project. The use of the double diamond in the development of the project is the iteration aspect of it as when something is built it is tested on users to collect their feedback then iterate based on the feedback until reaching the final version of the product.

In this chapter we will take a look on how the Data Science Portal was built from early stages until it reached its final version and released.

7.1. Project kick-off and team selection

Before building any products or adding any features it is important to identify the roles needed for this project and identify the team that will be responsible of the creation of the product and its delivery to the stakeholders.

The needed roles that were identified for this project are as the following:

- Product owner
- UX designer
- Graphic designer
- Front-end developer
- Content manager

Product owner

the product owner role in this project can be summarized as the following:

- Lead the effort from start to end
- Select the rest of the team members who will work on the project
- Figure out team members allocation percentage to the project
- Create the plan and the timeline for the project execution
- Arrange the weekly sync meetings among team members to review the progress and plan for the next week
- Communicate the progress to the stakeholders and get continuous feedback from the business
- Assist other team members when they face any sort of difficulties that prevents them from completing their tasks

UX designer

The role of the UX designer in this project can be summarized as the following:

- Design the user experience from the start to end.
- Design the functionality and workflow of the product from a user perspective and needs
- Collaborate with the graphic designer to implement his vision
- Create the scripts for the user feedback sessions
- Design the change based on user feedback

Graphic Designer

the main role that the graphic designer played in this project was to implement the vision and design of the UX designer and design the visual aspect of the UX designer's vision. The graphic designer was the one responsible for choosing all the colours, font type and size and every visual aspect of the product. Also, the graphic designer was the one to design the product banner choosing its colours size and location. The last part of the graphic designer's role was to communicate the design to the front-end developer who will code the design.

Front-end developer

The role that the front-end developer played in this project was to translate the graphic designer's work into a code using HTML, JavaScript, and creating the CSS files to implement the work that was done by the graphic designer.

Content manager

the product that this project will build is a knowledge base and the main aspect of any knowledgebase is its content. Without the content there is no product regardless of how beautiful or easy to use it is, the content or as we refer to it in this project the articles are the part of the project that brings value to the users out of using the product.

The role that the content manager played in the project can be summarized as the following:

- Help facilitate the workshop with data scientists to find what questions are relevant to them to include as articles in the product.
- Research the internal platforms like confluence and SharePoint to find out if there are already available content that can be used in the product.
- Design the article template.
- Identify the subject matter experts in the company to find the answers for the data scientists questions collected from the questions' discovery workshop
- Work with subject matter experts to create the articles
- Run the user feedback sessions
- Teach the subject matter experts how to create their own content and how use confluence to guarantee the content stays UpToDate
- Create the process of article creation
- Be responsible for all the content that will be published in the knowledge base to make sure the integrity of the information published that it is accurate without any incorrect or out of date information.

After the team was selected and roles were established the team arranged two weekly meetings to plan keep track of the project progress and divide weekly tasks then check tasks progress to make sure that the project is progressing, and the product is being implemented according to the plan.

Once the team was up and running it was time to know what should be the content that the knowledge base. In order to do so it was necessary not assume what type of content should be featured to eliminate any bias. Therefore, it was necessary to hold a discovery session with the target audience (the data scientists) to know from them directly what are the questions they would like to know the answer for.

7.2. UX discovery workshop to find out relevant content

Before starting to plan and design the workshop, the team first task was to create the categories that the product will be divided into. And the decision is to categorize the product

from a project a project lifecycle perspective, starting from business understanding until delivery and support (the categories were explained in chapter 4). The idea behind this sort of categorization is that every data scientist in the company is part of an ongoing project therefore if a data scientist needs assistance and visits the knowledge base it will be easier to navigate directly to the category that represent the phase of the project he/she is on and that would make it faster to locate the article guide that answers their question.

The main idea behind this workshop is to collect as much questions from data scientist as possible in order to have enough possible content for the product that we are building and since data scientists are in three different company locations, therefore three workshops were held to collect questions from data scientists in each location.

The format of the workshop was designed to give the freedom for the data scientists to provide all questions that they think is related to their work and any question related to the company as well to cover as much as possible.

The goal of the workshops is:

- Collect and prioritize most pressing questions data scientists need to know for their work (especially when they started working in company)
- Review and collect feedback on the high-level categories of product
- Review and prioritize desired functionalities of the Data Science Portal

Location 1 workshop

The focus group for this workshop is 10 data scientists with varying range of knowledge and company experience, from recently joined data scientists (less than 3 months), data scientists with more than 2 years' experience in the company, and senior data scientists with more than 4 years' experience in the company.

The workshop begins with giving everyone in the room sticky notes and markers, explaining to them that the goal of the workshop and then ask them to start writing questions and sticking them to the board.

After the collection of all questions from the data scientists, the next task was to know what are the priority questions that needs be worked on first. Therefore, every data scientist is provided with 5 small stickers and asked them to distribute the stickers on the questions they think are the most important with the possibility to stick more than one sticker on the same question if the data scientist deems it very important.

The last task for the was to arrange the questions in their respected categories by taking the sticky that has the question and then stick it under the correct category that is drawn on the white board.

After the workshop all questions with their importance score is documented in the project backlog on confluence.

Questions collected from the workshop per category:

Demand management

- How does demand management work? (1 point)
- Who are the CSLs for specific area/division?

Business understanding

- Who is who? Team structure across company/divisions/orgs (1 point)
- Who has specific domain knowledge? (1 point)
- Type of projects, self-service or standard process?
- How can I learn more about business domain?

Data acquisition and access

- How do I share data (large & small)? (2 points)
- How do I find data sensitivity? (1 point)
- How to create database? Where to put test data and create rules? (1 point)
- How can I get a database?
- How do I get access to different tools?
- How to work with data from 3rd parties?
- How to export data from platforms and whom to contact?
- How to make direct connection to database?

Setting up analytics environment

- How do I set up R/Python environment? Install packages, obtain R, IDE,... (1 point)
- How do I access GPUs?
- How do I get a licence to a specific tool? Who maintains them?
- Is there a high-performance computation environment? How do I access it and use it?
- Should I use cloud solution and which one?
- How do I set up cloud environment?

Pre-processing and modelling

- What APIs are available for pre-processing of data (1 point)
- Who can help me with specific kind of model? (1 point)
- Tools used for building mathematical models
- Tools for building data relationship models

Collaboration and application development

- Git - how do I create a project; how do I share access rights? (1 point)
- Documentation - where and how?
- What is the procedure for cooperating with different platforms?
- How to solve versioning capability issues?
- Where to keep all documents/notes/presentations for the project?

Visualization and dashboards

- What are the BI tools used for visualized results? (2 points)
- How to build visual reports and dashboards?
- Security models for different tools

Industrialization and productization

- How do transition project to maintenance? (1 point)
- What tool options do I have for productization? (1 point)
- How do I manage access rights to the model?
- How do I expose my model over API?
- How do I deploy my model?

Continuous monitoring

- How to support customer after delivery (1 point)
- What is the split of responsibilities in the support?
- How to integrate usage analytics?

Data science community and learning

- Who is who in Data Science team? (3 points)
- What are the past successful Data Science projects? (1 point)
- Capabilities and expertise inside Data Science team (technical and business domain) (1 point)
- What are other DS teams existing in MSD?
- How can I get literature? Books, journals & data bases
- What seminars do exist here (internal)?
- Responsibilities and process (each step) during the project
- How to take external trainings?

Location 2 workshop

The workshop in this location took the same approach of the workshop done in location 1 with focus group of 8 participants with mostly recently joined data scientists with less than a year experience in the company and 2 senior data scientists. The workshop was facilitated over a video conference and rather than sticky notes the data scientists were provided with a webpage to add their questions directly and then each data scientist can add up to 5 stars in total to the questions they believe are the most important to them.

Questions collected from the workshop per category:

Demand management

- How do we get projects?
- How do we manage capacity of the projects? (too many/few people on the project) *
- How do I know which projects are currently running in Data Science team? *

Business understanding

- No questions provided in this category

Data acquisition and access

- What kind of data sources are available in the company? ***
- Overview of most generic data sources available in the company?
- How do we get authorization to access the data (whom to contact)?
- How do we get external (outside of MSD) data sets into MSD?

Setting up analytics environment

- How do you install packages into Python/R? (sci-kit learn, ...) which are not yet available inside the company*
- How to choose the best (most efficient) algorithm for the specific task?
- What technologies are most commonly used by Data Science teams? **
- Where and how can I run a hosted version of R? and Python? **
- What high computing resources are available in the company? For which tasks to use which? *****
- Sandbox
 - How can I get a sandbox to try something? *
 - What is a difference between sandbox and production environment?

- Where can I try a new tool? *
- How do I manage access control for visualisations and data? *
- How do I build a project which needs some data to be updated manually by user (e.g. new price every month)?
- Is there any internal API which we can use during the project? **

Pre-processing and modelling

- Scheduling
 - How can I schedule a job to run at a specific time automatically? *
 - How do I create an automated pipeline of different tools used in a project? ***
 - Where can I store intermediate results during computation?

Collaboration and application development

- Where can I find code for specific project? ***
- Where do I keep my code?
- Where can I find data set used for modelling in a past project?
- Where do I archive my project data when project is done? *
- Where should I put all the project data (documents, presentations, ...) related to a project?
- How do I collaborate with other data scientists on a project? (how to use JIRA/Confluence/...)
- Stash:
 - How to use git from company command line - in an MSD specific tutorial?
 - How do I create an account in bitbucket?
 - How do I create a project in bitbucket?
- How do you do SDLC for Data Science project? ***
 - Are there any templates/guidance?
 - How do you use those templates?
 - What do I need for PoC, what do I need to do for industrialization?

Visualization and dashboards

- Is there any template to use for web development (CSS file, ...)? *
- What is the API to authenticate the users? And how to use it?
- How to put something on R Shiny?
- What visualisation tools are available?
- How do I connect Spotfire to Hive?

Industrialization and productization

- How do I transition a project to maintenance and support?
- How do I estimate cost of the project at the early stages of a project?
- How much does it cost to support the project on a specific solution/platform?
- How do I market our product (or successful PoC) to people in different divisions for similar use-case? **

Continuous monitoring

- How do I make sure that the Shiny dashboard is up? How do I register my Shiny dashboard and monitoring?
- Do we have any procedures for continuous monitoring? Is there a way to track errors and application events? (model is ready/run successfully/...)

Data science community and learning

- How do you find colleague with specific technology knowledge or who did similar work previously? **
- How is the Data science team organized? (by technology domain or business domain?)
- Are there any internal learning resources/courses?
- Is there any subscription to any database of scientific papers? *

For the third location, the workshop was done by colleagues from there without video conferencing due to time zone difference and it was concluded that there were no new questions asked and all the questions gathered from there were covered by the two workshops.

After gathering all the questions and combing them together removing the duplicate ones asked in different locations and the questions with the highest points or stars were highlighted, all the questions were added to Jira for tracking the progress.

The next step was to start researching the internal platforms to find which questions have answers already exists in order to include it in the portal. For questions that doesn't have answers documented in any internal system a research was done internally to locate the subject matter experts in the company for each question or area to help in creating an article within the article creation process.

7.3. Article creation process

In order to have content on the Portal there are two options, either the answer to the question exist on platforms like Confluence and SharePoint and in that case the person who wrote that answer is contacted to validate that what is written is still accurate and valid, or the question does not have any answer documented on any internal platform and in that case an expert is identified within the company and contacted to help create an answer and for that person will be the article owner for that question (see Figure 21).

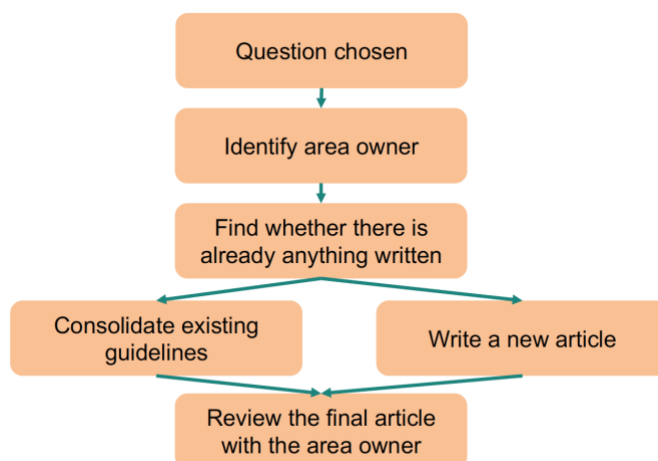


Figure 21: Article creation process

The process in Figure 21 describes the first process on how the content was created for all articles in the Data Science Portal. The reason for this process is to assure the integrity of whatever is published on the portal to be true and valid with an expert as an article owner in case users have more questions to follow up or would like to learn more from the experts

themselves. At the end of the day the portal is not only about the knowledge share rather it is about building a community for the data scientists and encourage a culture of learning and discussing among all data scientists regardless of their experience or location.

After the release of the Data Science Portal the process had minor changes for the purpose of continuous improvement and to be more self-explanatory for the librarian who is operating the portal after release and delivery (see Figure 22).

The need for process improvement is that after the release questions are not chosen anymore as the first release of the portal covered all answerable questions from the UX discovery workshops with the data scientists. The other channels that help create more content and articles are the functionalities of the portal:

- Create an article
- Request an article

Note: Both functionalities were covered in chapter 4.



Figure 22: New article creation process

7.4. UI design in confluence environment

From a graphic design perspective, no special measurements were needed for designing for confluence, but the implementation of the design required to learning the id of few of confluences elements in order to use them by coding rather than using those elements from the interface in confluence.

The main idea with the UI design is to change the perception of users when they visit the portal by making it look like an independent site that doesn't resemble a normal confluence workspace. To do so several tweaks to confluence were necessary in the CSS file including:

- Hiding the side bar

```
.ia-splitter-left,  
div.ia-fixed-sidebar {  
    display: hidden !important;  
    width: 0px !important;  
}
```

- Hiding the tools bar on the top of confluence

```
.space-tools-section {
    visibility: hidden;
}
```

- Hiding Metadata (confluence pages tree navigation) and Breadcrumbs (access to page visit statistics to users)

```
.with-breadcrumbs {
    display: none
}
.page-metadata {
    display: none}
```

- Altering the confluence space search design

```
fieldset.search-macro-fields {
    border: 20px solid #eff3f6 !important;
}
```

- Overriding confluence default settings for images being clickable to prevent the home page banner from being clickable by users

```
.confluence-embedded-image {
    pointer-events: none;
}
```

- Hiding the title of the homepage

```
/* hide menu and titles */
#title-text,
.page-metadata,
#likes-and-labels-container {
    visibility: hidden;
    display: none;
}
```

The design took several iterations based on the feedback collected from users (discussed in the next section).

Note: design iterations and can be found in the appendix B at the end of this paper.

7.5. User feedback – collection and analysis

After implementing the design and following the process of creating articles and filling the portal with enough content to have the first version of it ready, user acceptance testing is in order to get the sense of how users feel about the portal and to get what works and what needs improvements. In order to do so it is important to have a clear objective for the feedback sessions, note a hypothesis on how the users will perceive the portal, and create a script for the for running the user feedback sessions.

Objectives

Learn if following capabilities of DS Portal are understood and easy to use for data scientists:

- Finding requested information (search / browse)
- Navigating through the Data Science Portal
- Contacting the Data Science Portal team

Hypothesis

Home Page

- Users should divide 50/50 % into two groups in regard to preference of searching the information (search vs browse).
- Search is a quick and easy way to find information.
- Breakdown into categories for browsing makes sense for data scientists, when they search for a specific answer.

Category

- It's easy to locate requested information in the relevant area on Category page.
- Users will scroll the page containing more than 3 areas

Feedback

- "Feedback" plugin button would be used for sharing feedback with DS Portal team

Navigation

- For users it's easy to get to the page they want (navigation between category pages and back to home page)

Feedback session script

Objective	Script
Introduction	<ul style="list-style-type: none">• Welcome and thank you for coming• Purpose of session: Here to gather some feedback on the Data Science Portal, it's on the early content collection stage, so not all materials are available• There are no right or wrong answers and I would like to hear your honest opinions. That being said I would like to ask you to speak openly about what you like or what you dislike• It is also important to say that from now on your name is not going to be mentioned anywhere. The outcomes are going to be fully anonymized• Do you have any questions before we begin?
Finding information on the Home Page and Category page	<p><i>Show the home page</i></p> <ul style="list-style-type: none">• What do you think are you see?• Is there anything missing for you?• Imagine you need to find information about how to get software licenses <p><i>Observe if they use search or start browsing. Give a hint if to go to Category...</i></p>
Category	<p><i>If they used search, ask to navigate to Category "Setting Up Analytics Environment" and find answer there</i></p>

Navigation	You need to find an answer to another question what do you do?
Feedback	You found a mistake or an issue with the article, what do you do?
Wrap up	<ul style="list-style-type: none"> • General feedback • Imagine this portal is released tomorrow. How would you use it? • Anything missing for you? • Any suggestions?

Table 1: User feedback session script

The feedback sessions are held with maximum 2 team members of the data science portal so the data scientist participating in the session wouldn't feel uncomfortable and therefore feel pressured to give positive answers. The first team member is the facilitator of the session, the one who is interviewing the data scientist, and the second team member is setting in the corner of the room documenting the session without interfering in anyway.

The feedback session row results

There were 6 feedback sessions conducted with 6 different users with different roles and experience in the data science teams.

User 1

Background

- 2 months in the company
- Junior data scientist

Home page first impression

- She thinks that it should be for starting DS
- Like the general appearance of the portal
- Understands that on left is content, and the right column of the main page is for collaboration
- Little bit lost in the structure of the main categories, would preferer alphabetical order of categories
- She is missing the opening pitch: What is the purpose of this portal? Why is she here? Or what is this about?
- Using chrome to access the portal

Task 1. Find info Search for software license

- At first, she starts to scroll, cannot identify the main category
- Resorts to search
- Finds the correct article in recommended articles, clicks directly on it
- Starts gradually reading the article
- Feeling from article page:
 - Really appreciate problem definition, understands it as the main answer
 - Likes Article owner, would use him in case of additional questions

Task 2. Navigation Find another article

- Used back button to main page
- Clicks at category name to get to category page
- Quickly scans
- Goes back to main page for search

Task 3. Feedback button. If you found a mistake what would u do?

- Looks around and finds report an issue on right, uses it

Wrap up and conclusions

- Pretty pleased
- Interested in who will maintain the content
- Understands it as basic(starting) knowledge share

User 2

Background

- 2 months in MSD
- Junior DS
- Fresh graduate

Home page first impression

- Using chrome
- Knows about DS portal from Francois
- Spots that it is confluence page (asks about it to verify)
- Not clear about the title/purpose from the main page
- Prefers search (seems to be interested primarily in search field)
- Missing Title definition
- Understands/prefers present articles on the main page as recommended articles, which are must/nice to know

Task 1. Find info Search for software license

- Immediately starts typing into search
- Hits enter button
- Redirected
- After few seconds of struggling, the user picks the correct article
- Happy with article page structure

Task 2. Navigation Find another article

- Looks for search field on the page
- Immediately moves back to main page (using back button in browser)
- Types into search key words
- Does not completely understands information categorisation on the main page
- Does not know where R related stuff is

Task 3. Feedback button. If you found a mistake what would u do?

- Prefers to ask around in his office
- Then he would contact article owner how to get into contact with people responsible for DS portal

Wrap up and conclusions

- Very pleased
- Wants to start using ASAP
- Missing way how to contact/create new content
- Missing past projects in DS
 - His main wish is to be able to see previous projects with good documentation of processes and also easily approachable source codes, e.g. he want to just

copy them from the portal and immediately wants to start using the code and changing it to his needs

- Understand DS portal as experience sharing base
- Would use it also as a learning material in free time
- Go over links on main page and read them to enrich his knowledge

User 3

Background

- data analysis
- 3 months

Home page first impression

- clear
- looks like a webpage
- can see articles

Task 1. Find info Search for software license task

- how to get software license?
- using search got to the article

got to article page

- looks like Wikipedia
- clicked the same article in the related articles.

Task 2. Navigation Find another article

- Uses search to complete the task
- goes back to homepage clicking back.
- searches "install R"
- Asked to browse categories:
 - setting analytics environment
 - try to find the same article about licenses.
 - user uses ctrl F

Task 3. Feedback button. If you found a mistake, what would u do?

- use contacts and send an email to the contact or contact on jabber
- and on the article owner send email
- Question: How would contribute on the DS portal?
 - user doesn't know
 - tries to find comments but cannot or goes to someone from his team
 - user didn't notice the report an issue button

Wrap up and conclusions

- user trying to share the article using share from confluence. Understands hoe it works
- user saves page for later (Confluence button) and tries to find it on DS Portal.
- user didn't notice the feedback or report issue buttons
- user thinks report an issue is not about the article but more of a technical issue not article related and doesn't know who he is contacting or reporting for.

User 4

Background

- In company for 10 months
- Explanatory and predictive analytics

Home page first impression

- To find some data / visualisations
 - Led to think this because of the picture and header
- Cannot understand content based on the topic name
- Overwhelmed by amount of text
 - Too much text on landing page
- Confused by questions
 - Are there answers for them?
 - Should he personally answer them?
 - Why are there?
- Topic important and interesting for him personally
 - Data acquisition
 - Visualisations
- Does not understand importance/meaning of demand management

Task 1. Find info Search for software license

- Immediately uses search
- Identifies and clicks on proposed article

Task 2. Navigation Find another article

- Backspace to navigate inside portal
- Using IE search / searching for key words in category page

Task 3. Feedback button. If you found a mistake, what would u do?

- Proposes contacting article owner
- Considers it suboptimal
- Clicks on three dots on the top right corner ...
- JIRA like feeling
- Missing predefined information
 - Wants to see that the name of article is prefilled
- Does not understand what is meant by process
- Most likely would use article
- Thinks that it is more about technical part rather than proposing improvement
- Tries to find reporting on top right
- Doesn't find issue collector button

Wrap up and conclusions

- At the beginning of the session he thought it is about data
- At the end he thinks it is about processes
- Would need couple of positive experiences to start using it
- Setting up environment he prefers to do it by google
- Wants to know
 - How to classify data
 - How to do SDLC
 - Shiny app with HPC

- Where he can store/upload data of specific classification level
- Unsure about picture -> misleading
- Would appreciate special category for new hires

User 5

Background

- 2 months in the company
- Data scientist

Home page first impression

DS portal expectations

- Unification of multiple collaboration tools like Confluence/JIRA
- Unified structured project documentation
 - Quick overview
 - People working on it
 - Technology used
 - Links to final product/presentations
- Too big picture
- Too much text
- Not sure where to put his attention
- Overflow of information
- Finds questions redundant, would prefer it without them, just link to more topics
- Purpose according to him
 - Mixed feeling
 - Some categories for newbies
 - Other categories for more advanced users
 - Maybe managers only
- Unsure about some questions e.g. how to set up project funding
- Thinks that the questions are very specific to skilled persons who already knows what to do for starter, not sure how to start project at first

Task 1. Find info Search for software license

- Uses search
- Hits enter
- In search results confluence page retypes with different key word
 - List a lot more now
- Article feeling
 - Well structured
 - Appreciates problem definition

Task 2. Navigation Find another article

- Backspace to navigate inside portal
- More topics
- CATEGORY PAGE
 - Overwhelmed by amount of text
 - Navigates here by chrome search for keywords
- Ideally, he would have at least question per subtopic on main page
- Prefers different structure, would make some subcategories more visible e.g.
 - Python + R
 - HPC
 - Visualisations

Task 3. Feedback button. If you found a mistake, what would u do?

- Did not find Issue type for technical issues
- Start typing into drop down list
- Would expect it to be somewhere at top
- Would appreciate comment section either for proposing changes in the article or to improve its maintainability

Wrap up and conclusions

- Seems for him like categorized confluence
- He would use it at max through search
- No real incentive to currently use it
- He would make most viewed bigger with more than 5 questions
- He did not understand meaning of DS Community category
- Categories are useful for experienced users
- He did not see the frames on TV (e.g. problem definition, community on main page)
- He stressed out that HPC is very important topic, make it more visible and pay attention to content

User 6

Background

- One year in the company
- Project manager for data science projects

Home page first impression

- She would like to know what the page is for and who should use it.
- She expects there should be information on teams, capabilities, projects
- Also, she sees that there is technical information, she would expect to get contact information as well
- She often needs to explain to people what IT hubs are? and how could DS portal help them - and on the contrary, she would like to know how different organization in different countries work

Task 1. Find info Search for software license

- she straight away types it into the search bar - and clicks into article
- Article - she would prefer block alignment of the text
- She would appreciate short summary at the beginning in a form of a bullet list

Task 2. Navigation Find another article

- on a category page she is looking for a specific thing by browser search (ctrl+f) on a given page
- otherwise fine

Task 3. Feedback button. If you found a mistake, what would u do?

- She would contact Article owner
- she did not notice "report an issue" button - and if she did, she would think it is for technical problems on the page

Wrap up and conclusions

- She likes the category structure and names
- She would click on a "more articles" than the header of a category
- She likes the portal and thinks it is a good idea

Feedback summary

Home page

Findings:

Participants are missing a general message about the purpose and target audience of the portal (what it's about and who is it for).

Recommendation:

- Change the message in the top banner. Maybe add call-to-action using words "learn", "contribute", "discover"...
- Creating a distinct "About Us" page can help communicating purpose of portal and add more clarity on who's managing it and how people can contribute.
- Reduce height of the top banner according to some complaints.

Finding information

Findings:

- participants are using search instantly and can easily navigate to the article they are looking for
- browsing by categories might be used more for discovering new things by data scientists

Recommendation

- Keep in mind that users will be searching by keywords, when composing headlines of new articles.

Navigating the portal

Findings:

- Participants were mostly using browser or keyboard back button.
- Placing content in two rows on Category page (below the first scroll) didn't cause difficulties finding the article.
- Sometimes participants use browser search option (Ctrl+F on Windows)

Recommendation:

- Consider providing website-like navigation in top bar instead of Confluence navigation on Category and Article pages, so that users can easily get to Home Page.

Ability to find the feedback button

Findings:

- only one person found the Feedback button for reporting a mistake found in the article
- mostly participants would refer to article owner or leave comments

Recommendation:

- Having an "About Us" page could help explaining how the feedback channel works
- Turning on Confluence comments under articles could be helpful for users (not sure if it fits our strategy of maintaining the portal)
- JIRA modal is perceived as technical bug reporting tool. Renaming the it from "Report an Issue" to "Provide Feedback" or "Contact Us" should help.

The analysis of the feedback to come up with valuable recommendations to adjust the product to tailor it to the actual users' preferences was essential to form the final version of the product before releasing.

All recommendations were taking in consideration and the product was adjusted accordingly in preparation for the launch of the Data Science Portal.

Note: screenshots of the portal before user feedback and after with the final version for release can be found in the appendix of this thesis.

7.6. Launch of the Data Science Portal

After completing all the product adjustments for the Data Science Portal, it was time to think about the release of the product to users. The challenge is to make all data scientists aware of this product therefore marketing the product in the right form and make it easily accessible are very important to the success of the product because regardless of how good the product is and how much it is useful, if the target audience is not aware of its existence or if it was difficult to access.

To ensure awareness of the product among its target audience the following action plan was in motion:

- Creating an alias internal link for the confluence workspace that is hosting the Data Science Portal to make it easy to memorise for users capitalize on the company's culture of using this type of aliases
- Present the portal in the biggest internal data science event in the company that is attended by most data scientists in the company.
- Marketing by email to all the data science teams in the company informing them about the new product that was created for them with all necessary information about how to access and what can the product offer to them

Once the portal is accessible by everyone and it is up and running the portal was delivered to the stakeholders and a team from their side was trained to operate the portal and be the librarians to keep content up to date and help users add more content and to be responsible for the Jira project where the users feedback are gathered to take actions according to the user provided feedback on the portal using the feedback button.

8. Data Science Portal success evaluation and current status

The success of the Data Science Portal can be evaluated from different aspects, one aspect would be the number of visits to the portal by users, another aspects of success were not thought of or taken into consideration while creating this product but they came as a nice surprise to validate the effort put into creating the portal. These surprising aspects are:

- Providing feedback to different platforms on what is working and what needs some alteration
- Visible decrease in the number of questions related to how to do something from the data scientists to different IT teams in the company.

Data Science Portal visit statistics



Figure 23: Confluence view tracker for the Data Science Portal from Jan-Nov 2019

As it is clear in Figure 23 the portal had more than 17 thousand visits in the period from January to November 2019 which is a high number considering the target audience for this product is a little bit over a hundred data scientist, another important fact to look at from Figure 23 is the number of watches, it shows that there are 268 employees in the company that chose to subscribe to the Data Science Portal and to get notifications whenever there is a new article posted on the portal. The number of watches also shows that the reach of the portal is beyond the data scientists and other employees in the company are also interested in the knowledge that the portal provides, that is not surprising considering the fact that the portal provides articles on all aspects of projects and processes that can be beneficial for data scientists specifically and to all IT employees generally.

It is important to note the low number of comments and likes on the pages of the portal and explain that it is by design, the comment and like section that is by default on every page of Confluence was disabled on all the main pages of the portal such as the home page and category pages in order to guide the users to use the feedback button for their comments or

feedback, and the only place where the comment section was enabled is on the articles themselves to give more choices to users giving feedback on the articles either by the comment section or by using the feedback button available on every page of the portal.

Providing feedback to different platforms in the company

A lot of articles in the portal describes how a process is done for a specific platform and those platforms can get good feedback on their processes from the readers on what they think of these processes in order for them to stay connected with the users and to adjust their processes if the feedback they get is mostly negative. For example the R language user group is informed about the feedback users are giving about the articles related to R language about the process of installing R on a local environment and the negative feedback is communicated to the R language user group for them to take in consideration and they actually adjusted the process accordingly.

Decreased number of support questions to IT teams

Before the existence of the Data Science Portal, it was a common practice that data scientists would send emails to different IT teams in company asking them IT related questions in different areas regarding installation and setup of the tools they use and about company's processes in using different models etc.(see Figure 24) Since the portal is up and running the number of emails sent to IT teams have decreased to almost zero as all these kind of questions regarding tools are well covered in the portal's category "setting up analytics environment".

Hi [REDACTED]

In our team, we have developed a library in R, which we would like to submit to the global Open Source R community – CRAN, which is outside of our company. After talking to [REDACTED], he suggested that you might have some knowledge how to proceed with this action in line with our company policies. Could you please help us?

Thanks,

Figure 24: Example of questions sent to IT teams before the Data Science Portal

Data Science Portal today

The portal at launch was all created content by me and my team and it was mainly the answers to the questions collected from the data scientists in the workshops conducted in different company locations. At the launch of the portal the count of articles was 23 published articles with another 21 articles that were in creation at the moment of the launch of the portal.

Today the portal became part of the data science community in the company that data scientists contribute to it in order to grow it even bigger. By implementing the functionality that gives the data scientists to create new content with the create an article functionality in the portal, today the article hosts 83 articles in different categories describing different processes and providing enough guides to help the data scientists in their daily work and to help new data scientists in the company shorten their onboarding period by having this knowledgebase at their service to teach them all they need to know and help them get the knowledge they desire regarding data science to simply achieve the goal set in the beginning "Make data scientists life easier in the company".

The Data Science Portal is never a finished product with always something new to be added simply to achieve the slogan of the Data Science Portal "**Learn, share, improve our data science knowledge**" at the company.

9. Summary

The goal of this thesis is the creation of knowledgebase for the pharma company MSD to serve data scientists in the company's different company's locations around the world. To do so with minimum cost and build the most effective solution, design thinking and UX approach has been taken to make sure that the solution that is built is the right one and it provides the users with what they need in the easiest way possible.

Design thinking and user experience (UX)

Design thinking and UX date back to long time ago, to some extent some people say design thinking dates to ancient Greece, but the design thinking and UX we know today have evolved over the last century and was first noticed worldwide by 1995 by mainly Stanford design school and IDEO company, and the UX term being used for the first time by apple.

User experience is concerned with providing the users with the best experience while using a product and in order for designers to do so they need to follow the UX principles:

- UX is unavoidable
- You are not the user
- You compete with everything
- The user is on a journey
- Keep it simple
- Users collect experiences
- Speak the user's language
- Favour the familiar
- Focus on stability, reliability, and security
- Speed is essential
- Usefulness
- There are people in front of the interface

Following these principles assure that the product is user centric and would be a joy to the users and provide value to them.

On the other hand, design thinking is a process that works mainly to achieve to things, the first being identifying the right problem that needs to be solved without any prejudice, and then find the right idea that would solve the problem and productize it. The process of design thinking has 5 main phases:

- Understand: to do the research and talk to all stakeholders throughout discovery sessions to understand what are the problems that they are having
- Define: to define the right problem based on the research and discovery done in the first phase
- Ideate: to continue the discovery and gather as much ideas as possible from stakeholders (business, users, experts...)
- Prototype: creation of prototypes of the ideas from the previous phase that are feasible in terms of money, time, and business goals
- Test: test the prototype on users and get their feedback and iterate to any of the previous steps as much as needed until reaching the final prototype or product

The process of design thinking is iterative and there are different models of it applied in the real world by different companies and design schools. All the different models follow in one way or the other the same 5 phases described above.

The model that is being used for the purpose of this project is the double diamond model that was created by the design council institute. The double diamond consists of four steps:

- Discover: research and discovery to find the real problems and challenges (diverge)
- Define: finding the right problem that needs solving (converge)
- Develop: develop ideas that can solve the problem (diverge)
- Deliver: choose the right solution idea and create prototype, test it on users and iterate based on feedback (converge)

The double diamond end result is not necessary a finished product or a functioning prototype, depending on the context of the situation the end result can be a prototype and it also can be a sketch of what the product should be. It all depends on how long the design thinking workshop is allowed to be and what are the available resources.

The Data Science Portal

The data science portal is the result idea of applying the double diamond in the company MSD with regards to finding a solution that would offer data scientists more support and help newly hired data scientists to integrate with their teams and start contributing to work in a shorter period.

Applying the double diamond to find the right solution:

- Discover: the research was done among the different locations of the company around the world with all stakeholders from users to team leaders to the management of the global data science in the company. A lot of pain points and needs were gathered, and few examples are:
 - No visibility on what is currently available on the platforms for data scientists
 - Very hard for new hires to integrate - overwhelmed with the enterprise, platforms, complex infrastructure
 - Hard to choose the tools data scientists want to use per solution
 - No control on licenses and services bought by data scientists
 - No close collaboration between DS and IT working on their solutions
- Define: in this phase the focus was on the common problems and the problems that affect directly the target audience. Given the resources available and some problems being unfeasible to solve the focus was given to these problems:
 - No visibility on what is currently available on the platforms for data scientists
 - Very hard for new hires to integrate - overwhelmed with the enterprise, platforms, complex infrastructure
 - Hard to choose the tools data scientists want to use per solution
- Develop: developing the idea of what the solution can be, the problems are all of an information nature therefore the right solution would be a knowledge base but the ideas varied of what the right solution would be, and after applying a feasibility check the choice was to build a knowledge base internally capitalizing the company internal platform “Confluence”.
- Deliver: a step where the prototype is created, the knowledge base given the name Data Science Portal and a design prototype was created.

With that ending the design thinking process and starting of development of the product.

Building the Data Science Portal

The data science portal is built on top of Confluence platform which is a project management tool by the company Atlassian that allows users to customize their spaces using HTML, JavaScript, and CSS files.

To start building the portal, the most important part of it would be the information it hosts as a knowledgebase, therefore discovery sessions were held with data scientists in different location to collect from them all kind of questions that they have or have had when they joined the company in order to create articles that answers each question to fill the portal with knowledge that the data scientists care about and need.

The questions are categorized in 9 different categories that covers all project phases from business understanding to maintenance and support

Keeping the UX principles in mind, the creation of the portal began with forming a process for articles creation that would insure that the information in the article is true and up to date, designing the portal UI, and adding functionalities such as feedback, create an article, and request an article to insure that the user will have a great experience in the portal to learn, share, and improve the data science knowledge in the company.

After creating the first version of the portal with enough content, user testing feedback sessions are held to get what users think and how they feel about the product with a created feedback script that would have the users perform specific tasks on the page and also get their general feeling of the portal. From the feedback collected from the users the portal was adjusted to fit the feedback including changing the position of the feedback button from the right side to an actual button at the end of every page, creating a unique about us page to give users more information about the portal, and enlarging the search bar and changing its position as feedback proved that it is a feature that users depend on the most to look for content.

Launching the portal

The portal was launched in the biggest internal event of the year for data science in the company and presented to them with. Also, an advertising email was sent to all data scientists about the launch of the portal to let everyone know about it.

Success evaluation of the portal and current status

The statistics show that there are over 17000 visits to the portal in the year 2019 which is a good number considering that the target audience of the portal are a little bit over 100 data scientists. The portal also grasped the interest of other IT colleagues as the statistics shows that there are over 250 people are subscribed to the data science portal to get email notification whenever there is a new article posted, and also other IT team in the company have followed the footsteps of the portal and created their own knowledge base to different groups in the company copying the exact same approach and design of the Data Science Portal, or just seeing it and taking the idea of creating something meaningful knowledgebases using confluence rather than building an entire internal system from scratch.

The Data Science Portal that have started with 23 articles, today hosts over 80 articles mostly created by the data scientists themselves using the functionality of the portal that allows users to create new content before the Data Science Portal librarians make sure that the created content is valid and true by consulting with the subject matter experts in the company.

10. Conclusion

The conclusion and lessons learned from working on this paper and creating this project can be summarized as the following:

- Technology is not the answer to creating good useful product, technology is a tool available to everyone and it should be used in that context
- The means to creating useful products is to identify the right problem to solve which gives value to the product's users
- To identify the right problem to solve, it is important to follow a design thinking approach
- Design thinking can be implemented by anyone and it doesn't require a long experience to start applying to problem solving in companies and in real life
- Design thinking can help remove any biased ideas and perception and therefore focusing on identifying the real problems that a company is facing
- Products do not need to be technologically complicated to provide value to users as long as the product solves the real problem that the users are facing
- All ideas should be heard when it comes to solving problems and facing challenges, treating all ideas equally from all stakeholders including potential users, experts, and the business owners helps to find the solution to the right problem
- A product's success is not just about the problem it solves or about the goal that it helps its users achieve. It is also about the experience that the users have while using the product
- Users would abandon a great product that offers great functionality if the experience was not pleasant
- Putting users first and focusing on what they need and want is the key to a successful product
- Design thinking and UX are best utilized when they are combined together, design thinking to identify the right problem and the right solution, and UX to create a great experience for users that combines the goal of the product with ease of use and product that would leave good feelings to the users when they use it
- UX principles are the key to designing products that users would love and depend on
- User feedback and testing are essential to the success of a product
- User feedback should be done as soon as possible in an iterative process of design, build, test until reaching a product that users can rely on
- Investing in design thinking and UX is a must for creating great useful products

11. List of figures

Figure 1: UX connection between business and the user	10
Figure 2: Diffusion of innovation theory (LaMorte, 2019).....	13
Figure 3: Charting system with bad design	18
Figure 4 The Interface of Therac-2	19
Figure 5: Year in review shared by Eric Meyer	20
Figure 6: The sweet spot for ideas to prototype3	23
Figure 7: Design Thinking Process	24
Figure 8: Zurb old design thinking model (Zurb, Dsign Process, n.d.)	25
Figure 9: Zurb new design thinking model (Zurb, Dsign Process, n.d.).....	26
Figure 10: Double Diamond Model by design council (DesignCouncil, 2019)	27
Figure 11: IBM design thinking model (model, 2016)	29
Figure 12 Other design thinking models (Scott, 2019)	30
Figure 13 Double Diamond Model Phases (Whalen, 2019)	32
Figure 14: A simple Confluence workspace (Confluence, spacetree for confluence, n.d.)	38
Figure 15 Macro selection menu on Confluence.....	38
Figure 16: examples of macros used in an article	39
Figure 17: Creating an issue collector on Jira	41
Figure 18: General feedback prompt window Article feedback on the Data Science Portal and associated Jira ticket.....	42
Figure 19: Article feedback prompt & associated Jira ticket	42
Figure 20 Data Science Portal prototype sketch	49
Figure 21: Article creation process	56
Figure 22: New article creation process	57
Figure 23: Confluence view tracker for the Data Science Portal from Jan-Nov 2019.....	68
Figure 24: Example of questions sent to IT teams before the Data Science Portal	69

12. List of sources

1. Cambridge, U. o. (2013, January 8). *Financial Content: Cambridge University study states software bugs cost economy \$312 billion per year*. Retrieved from insights.jbs.cam.ac.uk: <https://insight.jbs.cam.ac.uk/2013/financial-content-cambridge-university-study-states-software-bugs-cost-economy-312-billion-per-year/>
2. Claveria, K. (2019, June 25). *13 stunning stats about product innovation*. Retrieved from visioncritical.com: <https://www.visioncritical.com/blog/stats-product-innovation>
3. Confluence. (2019). *Confluence features and functions*. Retrieved from confluence.atlassian.com: <https://confluence.atlassian.com/confeval/confluence-evaluator-resources/confluence-features-functions>
4. Confluence. (n.d.). *spacetree for confluence*. Retrieved from marketplace.atlassian.com: <https://marketplace.atlassian.com/apps/1210945/spacetree-for-confluence?hosting=server&tab=overview>
5. Council, D. (n.d.). *What is the framework for innovation*. Retrieved from designcouncil.ork.uk: <https://www.designcouncil.org.uk/news-opinion/what-framework-innovation-design-councils-evolved-double-diamond>
6. Araz, C. (2018, March 15). *Why you should ditch your UX definition, and use this one instead* . Retrieved from uxdesign.cc: <https://uxdesign.cc/we-have-lost-track-of-what-ux-actually-means-8d55259dadb0>
7. Atlassian. (2019). *Atlassian Home*. Retrieved from Atlassian.com: <https://www.atlassian.com/>
8. Basuel, G. (2016, October 19). *Roots of UX*. Retrieved from medium.com: <https://medium.com/@UXGavin/tree-roots-of-ux-3d932f4d156c>
9. DesignCouncil. (2019). *double diamond model*. Retrieved from designcouncil.org.uk: <https://www.designcouncil.org.uk/sites/default/files/asset/document/Double%20Diamond%20Model%202019.pdf>
10. DESIGN, T. R. (2014, October 21). *Jen Cohen Crompton*. Retrieved from aquent.com.au: <https://aquent.com.au/blog/user-experience-roi>
11. Dickerson, J. (2013, September 9). *Walt Disney: The World's First UX Designer*. Retrieved from uxmag.com: <https://uxmag.com/articles/walt-disney-the-worlds-first-ux-designer>

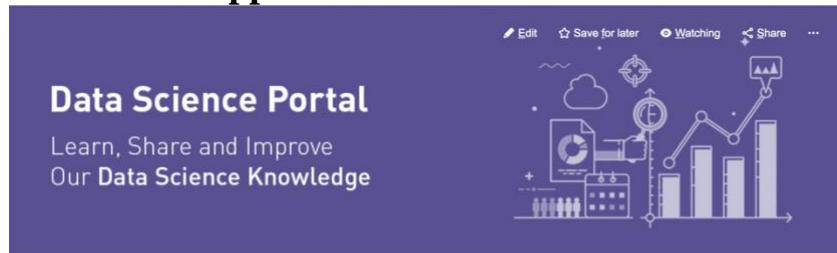
12. Hoffman, L. (2016, July 30). *10 Models for Design Thinking*. Retrieved from Medium.com: <https://medium.com/@elizabeth7hoffman/10-models-for-design-thinking-f6943e4ee068>
13. IBM. (2019, September 16). *enterprise Design Thinking*. Retrieved from IBM.com: <https://www.ibm.com/design/approach/design-thinking/>
14. Ingle, B. R. (2013). *Design Thinking for Entrepreneurs and Small Businesses: Putting the Power of Design to Work*. Apress.
15. Jewel, L. (2018, March 19). *Why businesses need to look at UX as a must-have investment*. Retrieved from Silicon republic: <https://www.siliconrepublic.com/companies/ux-design-business-benefits#targetText=It%20shouldn't%20come%20as,employees%20and%20reduced%20development%20costs>.
16. Jira. (2019). *Jira Software*. Retrieved from atlassian.com: <https://www.atlassian.com/software/jira/features>
17. Jira. (2019). *scrum boards*. Retrieved from atlassian.com: <https://www.atlassian.com/software/jira/scrum-boards>
18. LaMorte, W. W. (2019). *Diffusion of innovation theory*. Retrieved from sphweb.bumc.bu.edu: <http://sphweb.bumc.bu.edu/otlt/MPH-Modules/SB/BehavioralChangeTheories/BehavioralChangeTheories4.html>
19. Mailhot, J., Siminoff, E., Stanford, J., & Silverman, M. (2017). *What is design thinking*. O'Reilly Media, Inc.
20. model, I. (2016, January 2016). *IBM Design Thinking: A Framework To Help Teams Continuously Understand and Deliver*. Retrieved from IBM.com: https://www.ibm.com/blogs/think/wp-content/uploads/2016/11/lemniscate_Wired_invert-1.jpg
21. Modrono, T. (2018, November 19). *UX-O-METER # 01: Netflix vs HBO vs Prime Video*. Retrieved from Intractius.com: <https://blog.interactius.com/ux-o-meter-01-netflix-vs-hbo-vs-prime-video-54f125b9b6b>
22. MSD. (2019). *MSDIT*. Retrieved from msdit.cz: <https://www.msdit.cz/>
23. MSDIT. (2019). *Who we are*. Retrieved from MSDIT.CZ: <https://www.msdit.cz/about-us/>
24. Nieminen, J. (2018, October 10). *50+ statistics on innovation – What do the numbers tell us?* Retrieved from Viima.com: <https://www.viima.com/blog/innovation-stats>

25. Planet, U. (2017, July 25). *How To Calculate the ROI Of Your UX Activities*. Retrieved from UXplanet.org: <https://uxplanet.org/how-to-calculate-the-roi-of-your-ux-activities-b7ba7023246a#targetText=User%20experience%20is%20measurable%20in%20dollars%20and%20cents.&targetText=Knowing%20how%20to%20measure%20and,or%20on%20the%20CEO's%20mind>.
26. Planet, U. (2019, February 24). *What is UI design? What is UX design? UI vs UX: What's the difference*. Retrieved from uxplanet.org: <https://uxplanet.org/what-is-ui-vs-ux-design-and-the-difference-d9113f6612de>
27. Scott. (2019, May 2). *The different design thinking models- Unique way to position yourself to winning deals*. Retrieved from theunspokenpitch.com: <https://theunspokenpitch.com/deals/>
28. Saucier, c. S., & Shariat, J. (2017). *Tragic Design*. O'Reilly Media, Inc.
29. Soni, M. (2019). *DEFECT PREVENTION: REDUCING COSTS AND ENHANCING QUALITY*. Retrieved from isixsigma.com: <https://www.isixsigma.com/industries/software-it/defect-prevention-reducing-costs-and-enhancing-quality/>
30. Stull, E. (2018). *UX Fundamentals for Non-UX Professionals: User Experience Principles for Managers, Writers, Designers, and Developers* . Apress.
31. Szczepanska, J. (2017, January 4). *Design thinking origin story plus some of the people who made it all happe*. Retrieved from medium: <https://medium.com/@szczpanks/design-thinking-where-it-came-from-and-the-type-of-people-who-made-it-all-happen-dc3a05411e53>
32. UX, L. o. (2019). *Hicks Law*. Retrieved from lawsofux.com: <https://lawsofux.com/hicks-law>
33. w3schools. (2019). *CSS Tutorial* . Retrieved from w3schools.com: <https://www.w3schools.com/css/default.asp>
34. W3schools. (2019). *HTML intoduction*. Retrieved from w3schools.com: https://www.w3schools.com/html/html_intro.asp
35. Weir, S. (2019). *Our History*. Retrieved from designcouncil.org.uk: <https://www.designcouncil.org.uk/who-we-are/our-history>
36. Whalen, J. (2019). *Desgin for How People Think*. O'Reilly Media, inc.

37. Yeung, J. (2019, June 26). *New Stats Show HBO Has Retained Most of Its Subscribers Following 'GoT' Ending*. Retrieved from hypebeast.com:
<https://hypebeast.com/2019/6/hbo-subscriber-stats-post-game-of-thrones>
38. Zurb. (2019). *Our Story* . Retrieved from Zurb: <https://zurb.com/about>
39. Zurb. (n.d.). *Design Process*. Retrieved from zurb: <https://zurb.com/word/design-process#:~:targetText=In%20short,%20the%20design%20process,problems%20your%20product%20may%20have.>
40. Zurb. (n.d.). *Zurb Verify App*. Retrieved from Verifyapp: verifyapp.com

13. Appendix

13.1. Appendix A: The Data Science Portal screenshots



About

The Data Science Portal is a knowledge base intended for all data scientists in our company, curated by the Global IT Data Science Team, providing a single source of knowledge around internal processes. You can also share your tools and best practices here. The Data Science Portal will help you to get the most out of the wealth of Data Science information we share. Find out more about the Data Science Portal.

Browse by Category

Business Problem Identification

- People and team structure across company/divisions/orgs

[More articles](#)

Data Acquisition and Access

- What kind of data sources are available at MSD
- How to classify data
- How to find data sensitivity

[More articles](#)

Setting Up Analytics Environment

- How to set up a local R environment
- How to install packages into R
- Where and how to run a hosted version of R with HPC
- High performance computing resources at MSD

[More articles](#)

Data Analytics Trainings

Learn about and request hands-on training for analytics skills and tools available within our company.

Global IT Data Science Competency

- Literature, books, journals & subscription to databases of scientific papers
- Data Science Education Offerings

[More topics](#)

Have a Comment/Input?

[Give us Feedback](#)

Remember to add your name and email if you would like a reply from the team.

Preprocessing and Modeling

- How to get a licence to a specific tool/software
- What technologies are most commonly used by data scientists.
- Markov models for time series modelling
- Methods to model Loss of Exclusivity

[More articles](#)

Collaboration and Application Development

- Where to find source code for specific project
- Bitbucket (Git) - how to create a project and how to share access rights
- How to collaborate with other data scientists on a project
- Installing Emacs on Big Data Gateway machines

[More articles](#)

Visualization & Dashboards

- Templates for web development
- Available visualization tools and when to use them
- How to connect to hadoop cluster via spotfire analyst client
- How to connect Spotfire to Hive or Big Data lake

[More articles](#)

Industrialization / Productionization

- Where to find internal API and how / when to use them during

Operations and Support

- How to transition a project to AMS

Demand Management for IT Resources

- Getting the correct Client

Category example:

Note: All contacts details such as names and email addresses, and internal links are hidden from all screenshot

Setting Up Analytics Environment

Created by Raslan, Ahmad, last modified by

General Articles:

- [Accessing Admin Rights for Developers](#)

R Language

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It enables effective data handling and storage facility, a suite of operators for calculations on arrays, in particular matrices, a large, coherent, integrated collection of intermediate tools for data analysis and visualisations and many more.

Contacts:

Articles:

- [How to set up a local R environment](#)
- [How to install packages into R](#)
- [Where and how to run a hosted version of R with HPC](#)
- [How to run a hosted version of R with Hadoop](#)
- [Recommended style guide for R](#)
- [How to connect to and work with Oracle database from R](#)
- [Statistical Process Control package for R](#)
- [Running Parallel in R](#)

Useful links:

- [R User Group](#)
- [Tips for R](#)
- [R Software Infrastructure v2](#)
- [Yammer R group](#)

Python Language

Python is an interpreted high-level programming language for general-purpose programming. Lately, it is becoming more and more popular among data scientist, mainly for machine learning tasks and big data manipulation.

Contacts:

Articles:

- [How to properly set up Python environment](#)
- [How to manage conda environments and install Python packages into those environments](#)
- [Where to run a "hosted" version of Python](#)
- [Is there any standard style guide for Python](#)
- [How to install Python on your laptop \(for non-developers only\)](#)
- [Updating data with cx_Oracle](#)

Useful links:

- [Python \(@ MSD\) tips and tricks](#)
- [Python - Local development](#)
- [Python - Getting Started](#)
- [Python best practices](#)

Visualisations

Data visualisations make data more readable and comprehensible than the numbers or data tables because they employ visual channel of information input rather than textual. However due to data sensitivity rules you can learn here also how to protect and manage access for your visualisations.

Contacts:

Articles:

- [How to manage access control for data visualisations](#)
- [How to Install Spotfire Desktop Client](#)

Useful links:

- [How to show Google Analytics reports in a Confluence page?](#)
- [Data visualisation cheat-sheet, illustrated](#)
- [Data Visualisations Crash Course](#)
- [Data Visualisation Dashboard Wireframe](#)
- [Design a Data Visualisation in 5 minutes](#)

High Performance Computing

High Performance Computing helps to facilitate the rapid deployment and execution of complex or computationally intensive projects by managing high performance compute clusters, customised servers, workstations, high performance storage for research, and high performance networks.

Contacts:

Articles:

- [High performance computing resources at MSD](#)

Useful links:

- [High performance computing \(Sync\)](#)
- [HPC API usage](#)
- [HPC](#)

Can't Find the Article you are looking for?

[Request an article](#)

Industrialization / Productionization

Created by Raslan, Ahmad, last modified by I

SDLC - Software Development Life Cycle

The SDLC provides a single, comprehensive risk-based process model that governs the way IT solutions are planned, acquired, implemented, delivered, operated, maintained, and retired to ensure systems function as expected and are fit for intended use for the life of the system.

Contacts:

Articles:

- How to use SDLC for Data Science project
- Project Management - agile vs. waterfall structure for DS projects
- Using Spotfire Audit Documenter
- Special Considerations for Digital Health Solutions

Useful links:

- Model Lifecycle Management (MLM) Platform
- Easy SDLC
- Spotfire Audit Documenter

Architecture

When building more complex applications around analytics processes, you will probably need to integrate with several existing platforms and services. Architecture groups, especially ABRA and ARC can support you in that process by providing blueprints and best practices.

Contacts:

- ABRA team
- ARC team

Articles:

- Applying software engineering principles to your data science project
- MSD Analytics Platforms Comparison

Useful links:

- Team ABRA
- Architecture Review Crew (ARC)

API - Application Programming Interface

A powerful way to enable easy integrations and re-usability of analytics components is to expose them as a service. The two major platforms for publishing services are the API platform :

Contacts:

Articles:

- Where to find internal API and how / when to use them during the project
- How to expose model over an API
- Use Bash scripting with cURL to call API to obtain ISID from e.g. email address

Useful links:

Can't Find the Article you are looking for?

[Request an article](#)

About the Data Science Portal page screenshot

About

Created by Unknown User (krivanek), last modified by Raslan, Ahmad on Sep 26, 2019

What is the Data Science Portal?

The Data Science Portal is a knowledge base space intended for all data scientists in our company.

You can find all kind of answers to the questions that might rise during the course of your work. The answers are in the form of articles and step by step guides that you can read and learn how to perform a task or how a specific process works inside the company. The Data Science Portal is and never will be complete as there is more content added regularly.

Scope

- To provide data scientists with quick guidance on current processes and tools around analyzing data and deploying results
- Facilitate improvements across the data science tool stack and processes around it
- Drive discussion on future directions, tech evaluation, etc.

Vision

Make Data Scientists life easier in our company.

How to Contribute

For data scientists who would like to contribute to the data science portal there are several ways of doing so.

How to create an article

If you would like to add an article to the Data Science Portal click on the button below and start making your article. The article will be reviewed by the Data Science Portal team after saving for approval. It will be either approved and added to the Data Science Portal or we will send you an email with more questions and cooperate with you on creation of the article.

[Steps to Create an Article](#)

Request an article

Got a question on your mind that you cannot find the answer on the Data Science Portal?
Use the button below to request an article and we will get in touch with you.

[Request an article](#)

Give us feedback

On the Data Science Portal home page you can find the feedback button, we are eager to know what you think, what you like and what you don't like. please provide us with your feedback to help us improve.

Article feedback

On each article you can find article feedback button. You can give a feedback on the article itself if you think there is something wrong with it. Also you can give feedback on the process that the article describes and we will make sure that your voice is heard.

Article template

DS Portal Template

Created by Raslan, Ahmad, last modified by

After you publish the article it will be reviewed by the Data Science Portal team. It will be either approved and added to the Data Science Portal or we will send you an email with more questions and cooperate with you on creation of the article.



Problem definition

(in this section indicate the article's intended audience and the problem it solves or the process that it describes)

Article Owner



Raslan, Ahmad

(To be edited by the Data Science Team)

Solution overview

(in this section shortly explain the solution or the process but not in a detailed manner)

Related Articles

(To be edited by the Data Science Team)

Complete solution guide

(in this section write the complete solution in a step by step guide with all necessary details to make the reader apply the same steps to reach the solution or to reach a fuller understanding of the process)

Additional Information(if any)

(in this section add any links that could be useful and related to the topic of the article)

Note: leave this section empty if there is no links to add

Article feedback

the team)

(Note: Remember to add your name and email if you would like a reply from



Like

Be the first to like this

No labels



Write a comment...

Article example

| Data Science Portal | Setting Up Analytics Environment

EditSave for laterWatchingShare...

How to manage conda environments and install Python packages into those environments

Created by Raslan, Ahmad, last modified by

Problem definition

This article is intended for data scientists who uses Python language and want to learn how to manage conda environments and wants to install Python packages into it.

Solution overview

In this article you will learn the following :

- Create and manage conda environment
- Manage conda packages
- Manage non-conda packages

Complete solution guide

Managing conda environments is easy.

Best is to create `environment.yml` file and specify your dependencies on the beginning.

```
environment.yml
name: jupyter
dependencies:
  - python=3.6
  - jupyter
  - matplotlib
```

`conda env create -f environment.yml`
`conda activate jupyter`

Then later when you need additional package like pandas, you just add it to the environment.yml file like this:

```
environment.yml
name: jupyter
dependencies:
  - python=3.6
  - jupyter
  - matplotlib
  - pandas
```

`conda env update`
this will update your environment with pandas and its dependencies.

Managing non-conda packages

Conda has in repositories a lot of most used python packages. *But not all of them!* To also use PyPI and/or github/stash, you just specify it under `pip:` section in environment.yml, like this:

```
environment.yml
name: testing-flask
dependencies:
  - python=3.7
  - flask
  - pip:
    - Flask-Testing
    - docopt
    - schema
    - https://files.pythonhosted.org/packages/ac/8c/3c9f9be49ec299bbc5b563eb680
```

Article feedback

(Note: Remember to add your name and email if you would like a reply from the team)

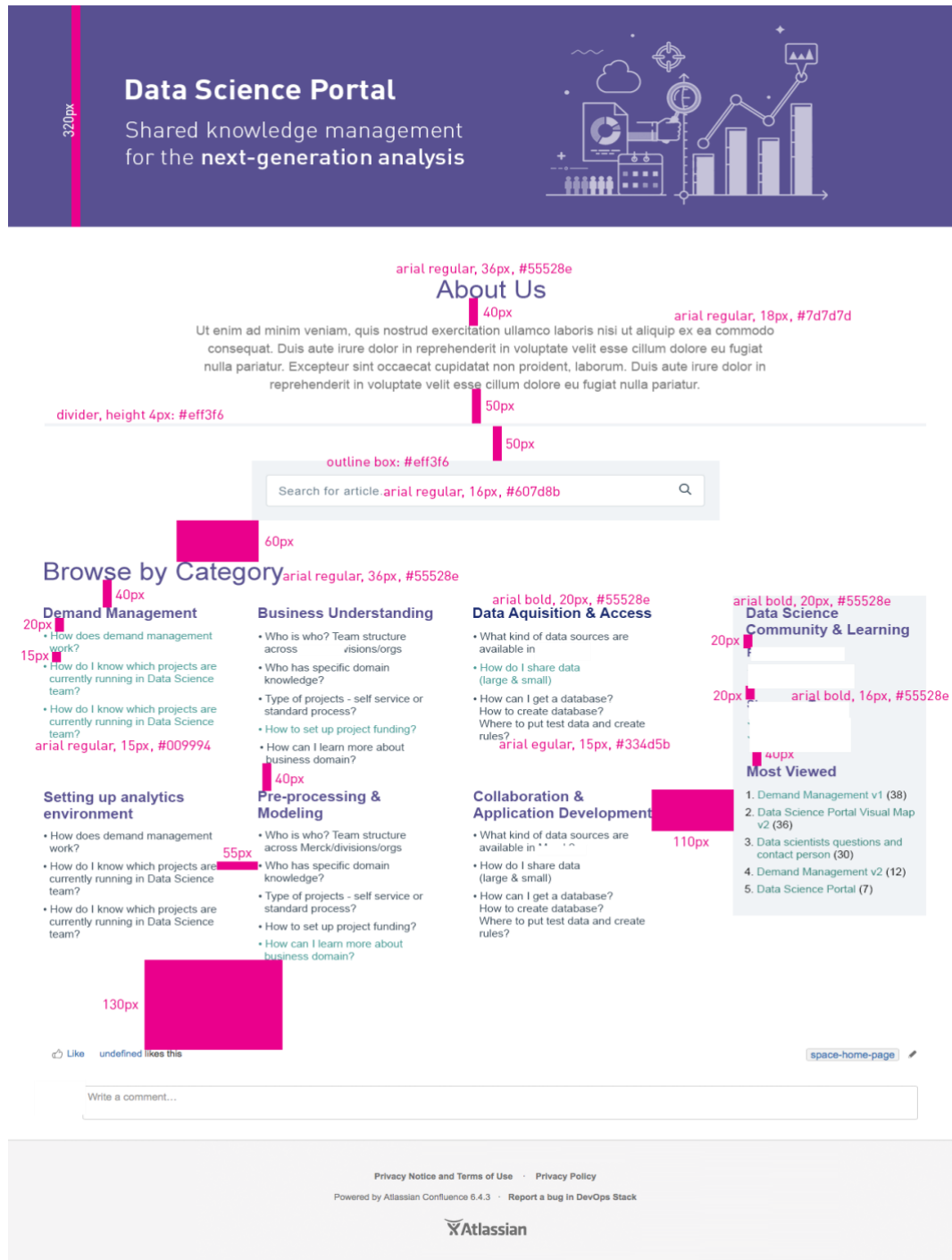
search

Article Owner

Related Articles

- How to set up a local R environment
- How to install packages into R
- Where and how to run a hosted version of R with HPC
- How to run a hosted version of R with Hadoop
- How to properly set up Python environment
- How to manage conda environments and install Python packages into those environments
- Where to run a "hosted" version of Python
- Is there any standard style guide for Python
- How to manage access control for data visualisations
- High performance computing resources at MSD
- Accessing Admin Rights for Developers
- How to install Python on your laptop (for non-developers only)
- Statistical Process Control package for R
- Recommended style guide for R
- How to Install Spotfire Desktop Client

13.2. Appendix B: Design iterations



Data Science Portal

Shared knowledge management
for the **next-generation analysis**



About Us

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, laborum. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.



Browse by Category

Demand Management

- How does demand management work?
- How do I know which projects are currently running in Data Science team?
- How do I know which projects are currently running in Data Science team?

Setting up analytics environment

- How does demand management work?
- How do I know which projects are currently running in Data Science team?
- How do I know which projects are currently running in Data Science team?

Business Understanding

- Who is who? Team structure across I /divisions/orgs
- Who has specific domain knowledge?
- Type of projects - self service or standard process?
- How to set up project funding?
- How can I learn more about business domain?

Pre-processing & Modeling

- Who is who? Team structure across Merck/divisions/orgs
- Who has specific domain knowledge?
- Type of projects - self service or standard process?
- How to set up project funding?
- How can I learn more about business domain?

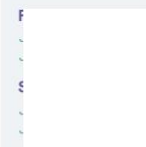
Data Aquisition & Access

- What kind of data sources are available in I
- How do I share data (large & small)
- How can I get a database? How to create database? Where to put test data and create rules?

Collaboration & Application Development

- What kind of data sources are available in I
- How do I share data (large & small)
- How can I get a database? How to create database? Where to put test data and create rules?

Data Science Community & Learning



Most Viewed

1. Demand Management v1 (38)
2. Data Science Portal Visual Map v2 (36)
3. Data scientists questions and contact person (30)
4. Demand Management v2 (12)
5. Data Science Portal (7)

Like undefined likes this

space-home-page

Write a comment...

Privacy Notice and Terms of Use · Privacy Policy

Powered by Atlassian Confluence 6.4.3 · Report a bug in DevOps Stack



Data Science Portal

Shared knowledge management
for the next-generation analysis

Learn & Contribute



Search for article...



Browse by Category

Demand Management

- How does demand management work?
- How do I know which projects are currently running in Data Science team?
- How do I know which projects are currently running in Data Science team?

Setting up analytics environment

- How does demand management work?
- How do I know which projects are currently running in Data Science team?
- How do I know which projects are currently running in Data Science team?

Business Understanding

- Who is who? Team structure across divisions/orgs
- Who has specific domain knowledge?
- Type of projects - self service or standard process?
- How to set up project funding?
- How can I learn more about business domain?

Pre-processing & Modeling

- Who is who? Team structure across Merck/divisions/orgs
- Who has specific domain knowledge?
- Type of projects - self service or standard process?
- How to set up project funding?
- How can I learn more about business domain?

Data Acquisition & Access

- What kind of data sources are available in ...
- How do I share data (large & small)
- How can I get a database? How to create database? Where to put test data and create rules?

Collaboration & Application Development

- What kind of data sources are available in ...
- How do I share data (large & small)
- How can I get a database? How to create database? Where to put test data and create rules?

Data Science Community & Learning



Most Viewed

1. Demand Management v1 (38)
2. Data Science Portal Visual Map v2 (36)
3. Data scientists questions and contact person (30)
4. Demand Management v2 (12)
5. Data Science Portal (7)



Like undefined likes this

space-home-page

Write a comment...

Privacy Notice and Terms of Use · Privacy Policy

Powered by Atlassian Confluence 6.4.3 · Report a bug in DevOps Stack

