## Appendix 1

## Peer Review Form

Date : 30/03/2022

Module Title: Carland Motors Assessment No: DWD\_506\_1 &\_2

My name: Vishwakranti Suryawanshi My partner’s name: James Lin & Judith Machingura

This form is intended to let you review the participation of your partner in this assignment.

Do not take into account technical skills.

Grading scale

You must grade your partner for each item listed in the tables below. 1 being the lowest, 5 the highest.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |
| Strongly disagree | Disagree | Ok | Agree | Strongly agree |

## Review your peer

|  |  |  |
| --- | --- | --- |
| Item | Grade | Comment |
| Completed equal work | 5 | Equally shared the work |
| Participated in decision making | 4 | Decision making capability is average |
| Kept you informed of the progress | 5 | Always updated about the website progress |
| Met timelines | 4 | Average |
| Responded promptly to problems | 4 | Quickly responded to resolve the problems |
| Met overall project objectives | 5 | We have completed all task promptly |
| Was open to new ideas and suggestions | 4 | We have shared and involved all ideas in the website development |
| Was easy to work with | 5 | Absolutely, we shared thoughts about different aspect of the project and executed it |

## Comments

Provide any extra comments on your partner’s contributions. If there was any conflict, explain how you handled it.

**Assessment brief Assessment - 1**

1. Design User-Interface, with mockups of all web pages for one screen size. Select one from desktop, tablet or mobile.
2. Produce a system design (Use-Case and Activity Diagrams) – Lucid chart for both designs Attached.
3. Plan a suitable relational database and present the data using an Excel sheet. This means no repeated information wherever possible and data divided into appropriate tables. Be sure to specify the correct data types, keys and indexes such as auto increment in your plan. – Excel sheet attached as per the specific mentioned.
4. Outline the visual design principles you considered and how you applied them.

* 5 visual-design principles that impact UX :

The principles of scale, visual hierarchy, balance, contrast, and Gestalt not only create beautiful designs, but also increase usability when applied correctly.

As per the principles, we have applied all in a proper manner.

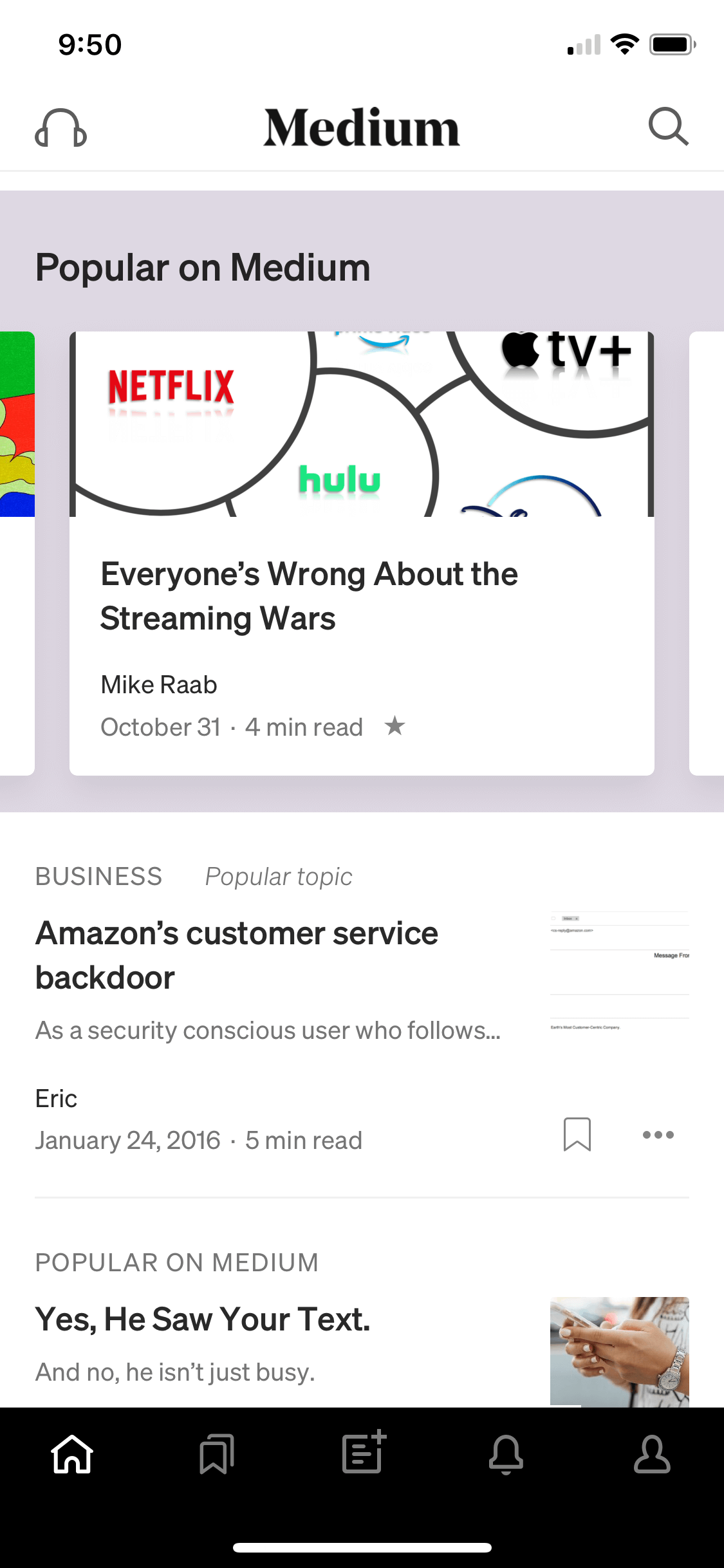
**Visual-design principles** inform us how design elements such as line, shape, color, grid, or space go together to create well-rounded and thoughtful visuals.

* Following these 5 visual-design principles can drive engagement and increase usability.

## Scale

When this principle is used properly, the most important elements in a design are bigger than the ones that are less important. The reason behind this principle is simple: when something is big, it’s more likely to be noticed.

* As per the scale principle in our website, we have kept the car photos and the titles in a large scale and more info car details links kept big.



*Medium for iPhone: Popular articles are visually larger than other articles. The scale directs users to potentially more-interesting article.*

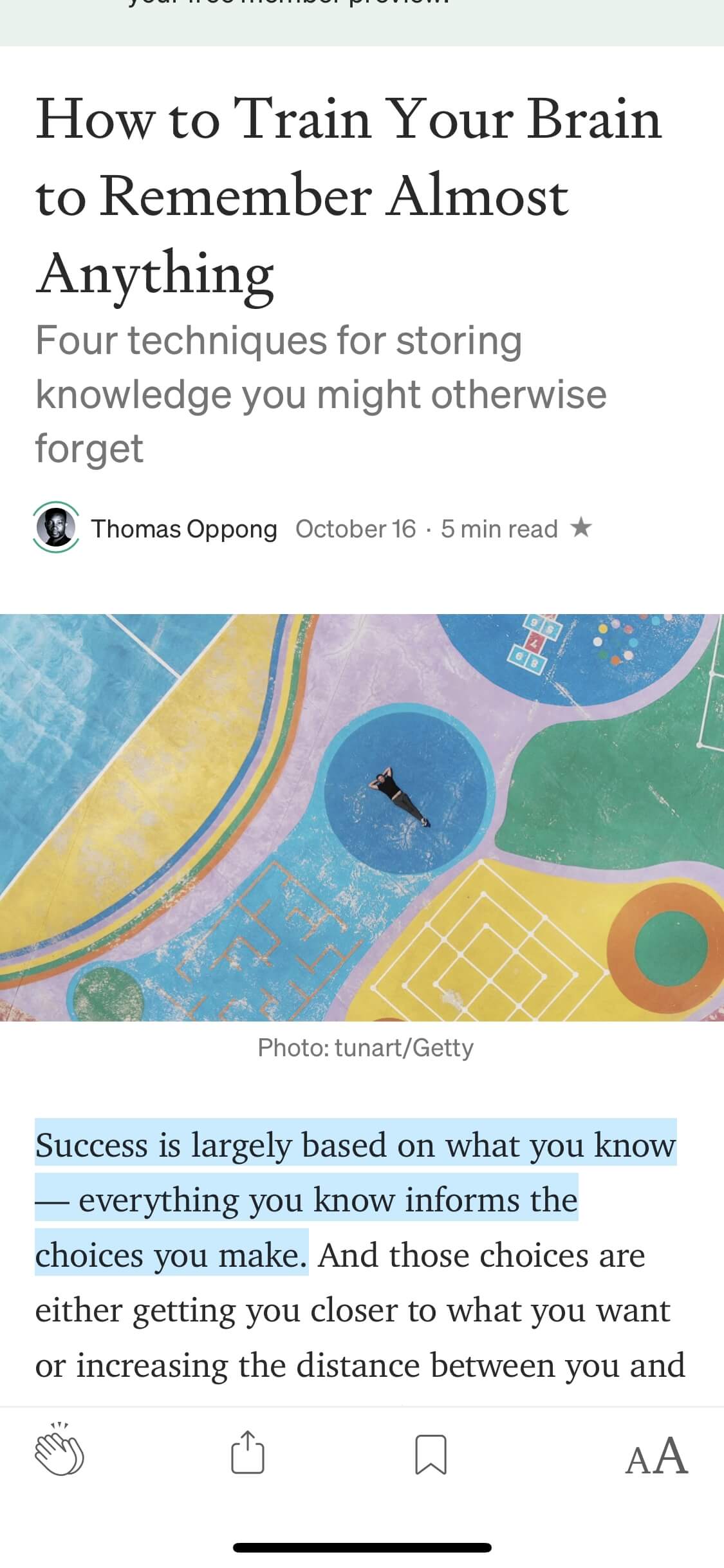
## Visual Hierarchy

* As per the principle we have used size and scale to emphasize visual hierarchy.
* We have designed very big text that explains all about the specific car.
* We also use “hierarchy” to show relationships between content blocks. Users define the visual hierarchy of a website or app.

A layout with a good visual hierarchy will be easily understood by your users.

The principle of **visual hierarchy** refers to guiding the eye on the page so that it attends to different design elements in the order of their importance.

To create a clear visual hierarchy, use 2–3 typeface sizes to indicate to users what pieces of content are most important or at the highest level in the page’s [mini information architecture](https://www.nngroup.com/articles/mini-ia-structuring-information/). Or, consider using bright colors for important items and muted colors for less important ones.



*Medium mobile app: There is a clear visual hierarchy of title, subtitle, and body text. Each component of the article is in a type size equal to its importance.*

## Balance

* We have balanced our website with the help of the distribution of the visual weight of objects, colors, texture, and space.
* We have used design in a proper scale manner, so these elements should be balanced to make a design feel stable.

Balance is like a seesaw: instead of weight, you are balancing design elements.

The principle of **balance** refers to a satisfying arrangement or proportion of design elements.

In a balanced design, no one area draws your eye so much that you can’t see the other areas (even though some elements might carry more visual weight and be focal points). Balance can be:

* Symmetrical: elements are symmetrically distributed relative to the central imaginary axis
* [Asymmetrical: elements are asymmetrically distributed relative to the central axis](https://www.nngroup.com/articles/visual-design-cheat-sheet/)
* Radial: elements radiate out from a central, common point in a circular direction.

The kind of balance you use in your visual depends on what you want to convey. Asymmetry is dynamic and engaging. It creates a sense of energy and movement. Symmetry is quiet and static. Radial balance will always lead the eye to the center of the composition.

Nike: This page is asymmetrically balanced, giving a sense of energy and movement that is fitting to Nike’s brand. If you were to draw a vertical axis down the center on this visual, the number of elements is about the same on both sides of the axis. However, the difference is that they are not identical and in the same exact locations. Even though there is technically a bit more text on the left side of the shoe, it is balanced out with the larger text on the right that takes up more space and visual weight, thus making them appear pretty similar.



Brathwaite wrist watch: This classic watch is balanced radially. The eye is immediately drawn to the center of the watch face and all visual weight is distributed equally, regardless of where the imaginary axis is drawn.

## **Contrast**

This is another commonly used principle that makes certain parts of your design stand out to your users.

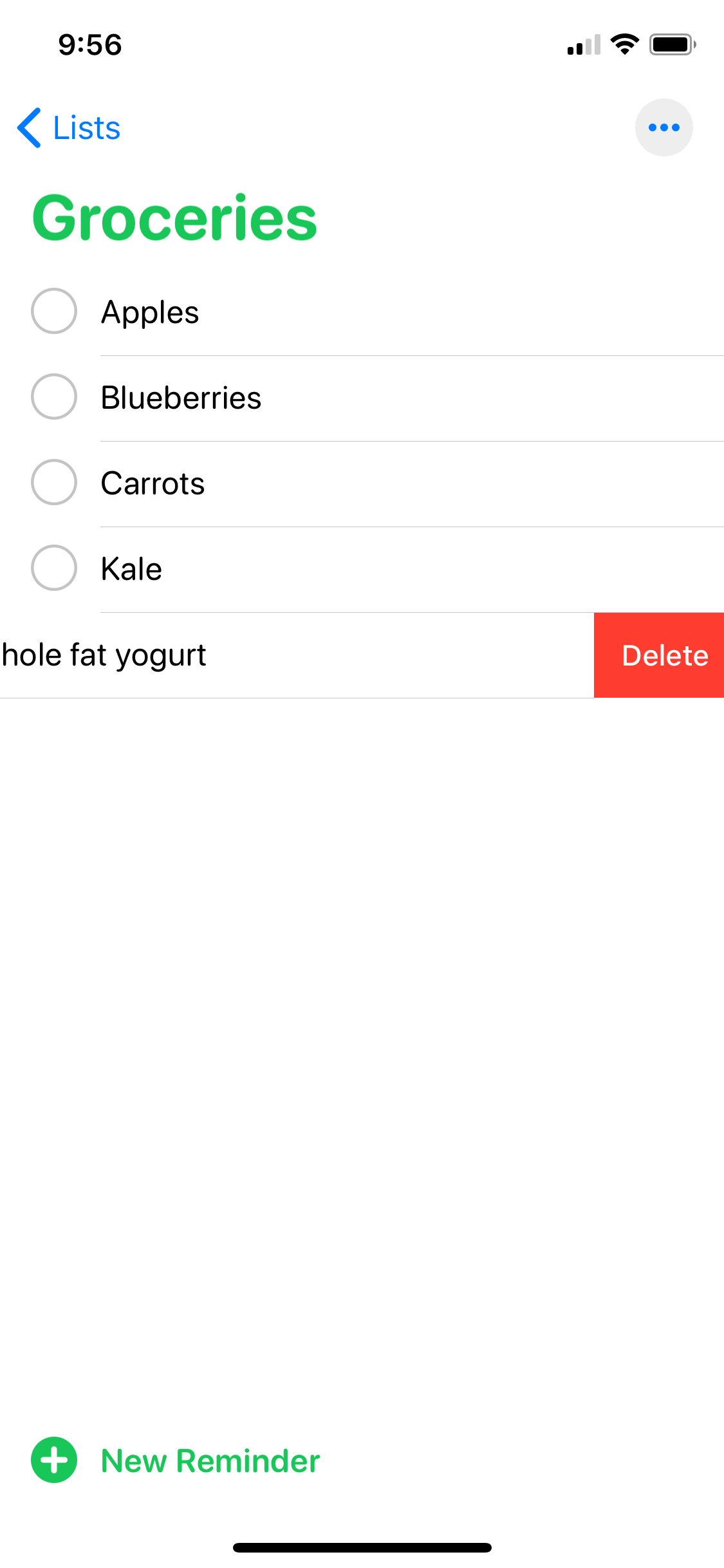
* We have used  the contrast between different elements on a website, whether it is colors, shape or size, is important to ensure that each element looks different than others.
* For example, light and dark colors, smooth and rough textures, large and small shapes.
* This not helps the user in differentiating between objects but also helps in readability and navigation on the website.

We have used 4 major types of contrast in web design:

* Light and dark.
* Colour.
* Foreground and background.
* Size and shape.

 Contrast provides the eye with a noticeable difference (e.g., in size or color) between two objects (or betwe`en two sets of objects) in order to emphasize that they are distinct.

The principle of contrast is often applied through color. For example, red is frequently used in UI designs, especially on iOS, to signify deleting. The bright color signals that a red element is different from the rest.



Reminders app on iOS: The color red, which has high contrast to its surrounding context, is reserved for deleting.

Often, in UX the word “contrast” brings to mind the contrast between text and its background. Sometimes designers deliberately decrease the text contrast in order to deemphasize less important text. But this approach is dangerous — [reducing text contrast](https://nngroup.com/articles/low-contrast/) also [reduces legibility](https://nngroup.com/articles/legibility-readability-comprehension/) and may make your content inaccessible. Use a color-contrast checker to ensure that your content can still be read by all your target users.



Greenhouse Juice Co: The legibility of the text on the bottle relies on the color of juice. Although the contrast works beautifully for some juices, labels for bottles with light colored juices are nearly impossible to read. (Image source: [*www.instagram.com*](https://www.instagram.com/p/BdqQjEIASuR/))

## **5. Gestalt Principles**

**Gestalt principles** explain how humans simplify and organize complex images that consist of many elements, by subconsciously arranging the parts into an organized system that creates a whole, rather than interpreting them as a series of disparate elements.

* We have covered in our website, five of the key Gestalt principles that can be applied in web design – proximity, similarity, continuation, symmetry and closure.

There are several Gestalt principles, including [similarity](https://www.nngroup.com/articles/gestalt-similarity/), continuation, closure, [proximity](https://www.nngroup.com/articles/gestalt-proximity/), [common region](https://www.nngroup.com/articles/common-region/), figure/ground, and symmetry and order. Proximity is especially important for UX — it refers to the fact that items that are [visually closer together](https://www.nngroup.com/articles/closeness-of-actions-and-objects-gui/) are perceived as part of the same group.



It is the Gestalt closure principle that allows us to see two figures kissing instead of random shapes in Picasso’s painting. Our brains fill in the missing pieces to create two figures.



We also often see applications of the Gestalt theory in logos. In the NBC logo, there is no peacock in the white space, but our brain understands there to be one.

## Why Visual-Design Principles are Important

* **Increase usability.** Following these visual-design principles often results in layouts that are easy to use. For example, the golden ratio, which is frequently used for creating beautiful works of art was also used in typesetting to create a visually pleasing relationship between font size, line height, and line width. The result typically led to shortened line lengths, which created balance (via white space) on a webpage and made the text easier to read. When paired with a strong interaction design, visual design will increase task success rates and user engagement.
* **Provoke**[**emotion**](https://www.nngroup.com/courses/persuasive-emotional-design/)**and delight.** Beautiful things elicit positive emotions. (In fact, the [aesthetic–usability](https://www.nngroup.com/articles/aesthetic-usability-effect/) effect says that when people find a design visually appealing, they may be more forgiving of minor usability mishaps.) By following  the principles of good visual designs, designers can create UIs that look good and thus make users feel good.
* **Strengthen brand perception.** A strong visual system builds user trust and interest in the product and appropriately represents and reinforces the [brand](https://www.nngroup.com/articles/brand-experience-ux/).

**Use of the software development methodology and explanation of 3 contemporary methodologies:**

Agile methodologies

Agile methodologies are approaches to product development that are aligned with the values ​​and principles described in the [Agile Manifesto](https://agilemanifesto.org/) for software development. Agile methodologies aim to deliver the right product, with incremental and frequent delivery of small chunks of functionality, through small cross-functional self-organizing teams, enabling frequent customer feedback and course correction as needed.

Application of Agile Methodology

Through most of its brief [history](https://en.wikipedia.org/wiki/Agile_software_development#History) (since 1999-2000), “Agile” has been predominantly an approach to software development and IT application development projects.  Since then, however, it now extends to other fields, too, especially in the knowledge and services industries.

Agile is about being responsive to the market and to the customer by responding quickly to their needs and demands and being able to change direction as the situation demands.  Be it IT or software development or any other field where there is a flow of work and delivery of work products, Agile methods are applicable.  Agile methods attempt to maximize the delivery of value to the customer and minimize the risk of building products that do not – or no longer – meet market or customer needs.

They do this by breaking up the traditionally long delivery cycle (typical of the legacy “waterfall methods”) into shorter periods, called sprints or iterations. The iteration provides the cadence for delivering a working product to the customer, getting feedback and making changes based on the feedback.

Thus, Agile methods have sought to reduce delivery times (delivering early, delivering often) to ensure that smaller vertical chunks of the product get to the market, enabling customers to provide feedback early and ensure that the product they finally get meets their needs.

Agile has become an umbrella term for a variety of planning, management and technical methods and processes for managing projects, developing software and other products and services in an iterative manner. These methods include Scrum, by far the most prevalent and popular method for software, XP (eXtreme Programming or Paired Programming), and more lately Kanban.

Agile methods also include technical practices – most of which fall under the umbrella term DevOps – that enable Test Automation, Continuous Integration/ Continuous Delivery/ Deployment (CI/ CD) and overall, an ever-shrinking delivery cycle for software and other products and services.

The use of Agile as an approach to project management has increased dramatically in recent years. [Gartner predicts](https://www.gartner.com/webinar/3169117) that agile development methods will soon be used in 80% of all software development projects.

The [Agile Manifesto](http://agilemanifesto.org/principles.html) is a statement of core values ​​and principles for software development. The Agile Manifesto for software development was set up in 2001 and it is a declaration of 4 vital rules and 12principles that serve as a guide for people in [agile software development](https://www.digite.com/swiftenterprise/). It was created by 17 professionals who already practiced agile methods such as XP, DSDM, SCRUM, FDD, etc, gathered in the snowy mountains of the US state of Utah, convened by [Kent Beck](https://www.kentbeck.com/)[.](https://www.digite.com/swiftenterprise/)

#### Core values of Agile Manifesto

Individuals and interactions over processes and tools – The first value emphasizes teamwork and communication. We must understand that software development is a human activity and that the quality of interaction between people is vital. Tools are an important part of software development, but making great software depends much more on teamwork, regardless of the tools team may use.

Working software over comprehensive documentation – Documentation has its place and can be a great resource or reference for users and coworkers alike. The main goal of software development, however, is to develop software that offers business benefits rather than extensive documentation.

Customer collaboration over contract negotiation – Development teams must work closely and communicate with their customers frequently. By listening to and getting feedback, teams will understand what all stakeholders really want.

Responding to change over following a plan – Changes are a reality in Software development, a reality that your Software process should reflect. A project plan must be flexible enough to change, as the situation demands.

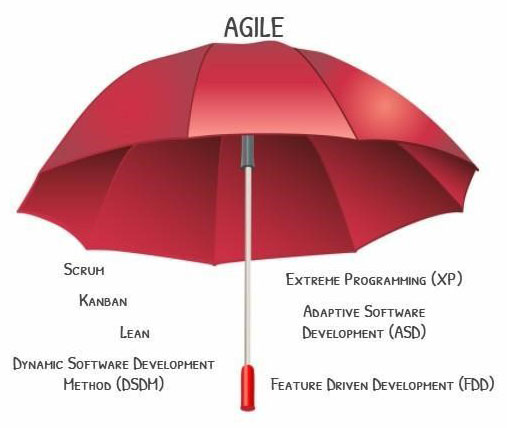
DevOps is the direct descendant of [agile software development](https://www.synopsys.com/glossary/what-is-agile-sdlc.html), born from the need to keep up with increased software development velocity and throughput agile methods. Advancements in agile development highlighted the need for a more holistic approach to the software delivery life cycle, resulting in DevOps.

“Agile development” is an umbrella term for several iterative software development methodologies, many of which have carried over to DevOps:

#### Key Agile Methodologies

Agile is an umbrella term for several methods and practices. Let’s look at some of the popular methodologies:

* Scrum
* Extreme Programming (XP)
* Adaptive Software Development (ASD)
* Dynamic Software Development Method (DSDM)
* Feature Driven Development (FDD)
* Kanban
* Behavior Driven Development (BDD)

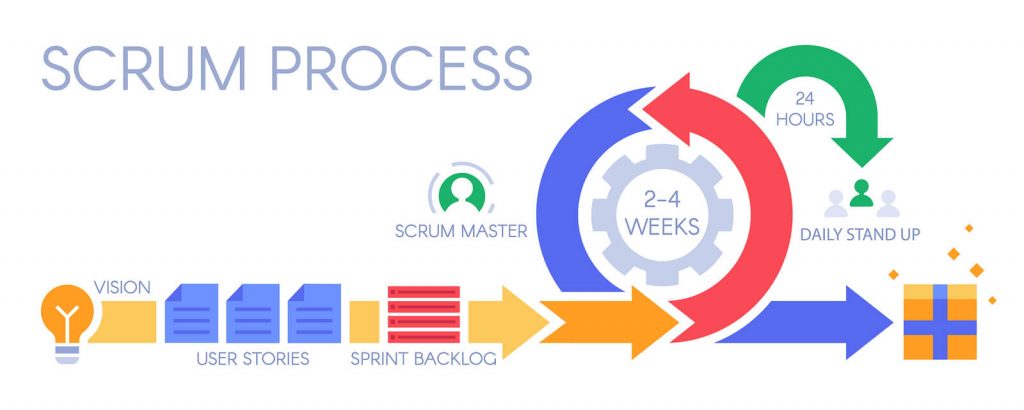


Scrum Methodology

Scrum methodology is a simple [framework](https://www.scrum.org/resources/what-is-scrum) for working with complex projects, and it was created by [Ken Schwaber](https://www.scrum.org/team/ken-schwaber) and [Jeff Sutherland](https://www.scrumalliance.org/community/profile/jsutherland).

Agile software development methodologies are iterative, meaning the work is divided into iterations, which are called Sprints in the case of Scrum. Scrum is executed by small teams of between 7-9 people, including a Scrum Master and a Product Owner.

In Scrum, projects are divided into cycles (typically 2 or 3 week cycles) called Sprints. The Sprint represents a timebox within which a set of features must be developed.  Multiple sprints might be combined to form a Release – where formal software/ product delivery is made to the customer/ market.



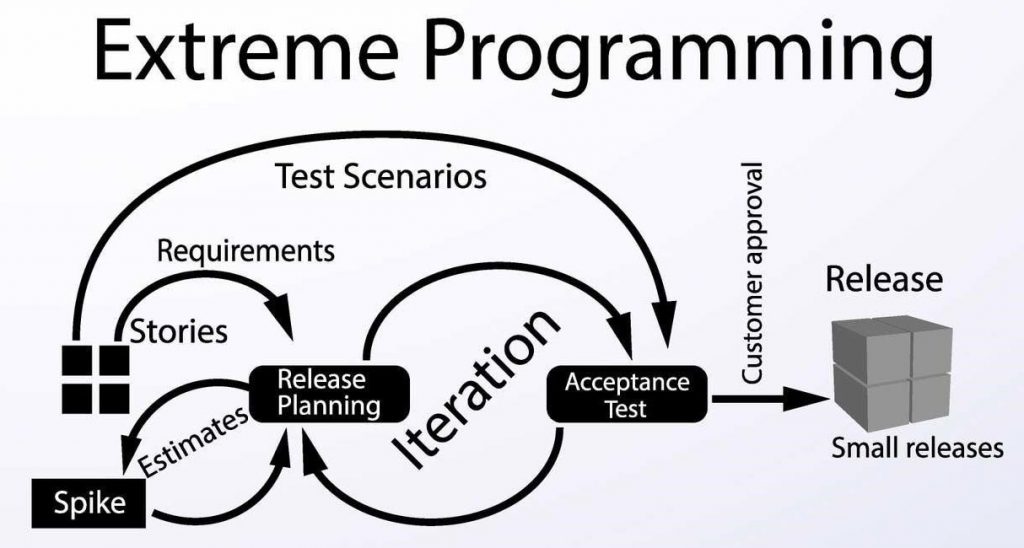
The overall product functionality is broken down by the Product Owner into smaller features (typically described as Epics and User Stories – or just Stories).   These Stories are prioritized and taken up in each Sprint or Iteration.  The intent of the method is for the team to be able to demo at the end of each Sprint working pieces of the product to the Product Owner, to make sure that the product is working as intended.

Overall, the Scrum method breaks the long waterfall process delivery into smaller cycles, which enables product teams and the end-customer to frequently review working software and ensure that it meets their business requirements. This ensures that the end product also meets the final requirements of the customer.

The Scrum method is characterized by specific ceremonies such as the Daily Standup meeting, the Sprint Review Meeting, the Demo to the Product Owner and the Sprint Retrospective meeting.  All of these meetings provide collaboration and review opportunities to the team to ensure that development is progressing as intended, and any issues are resolved quickly.

Extreme Programming (XP)

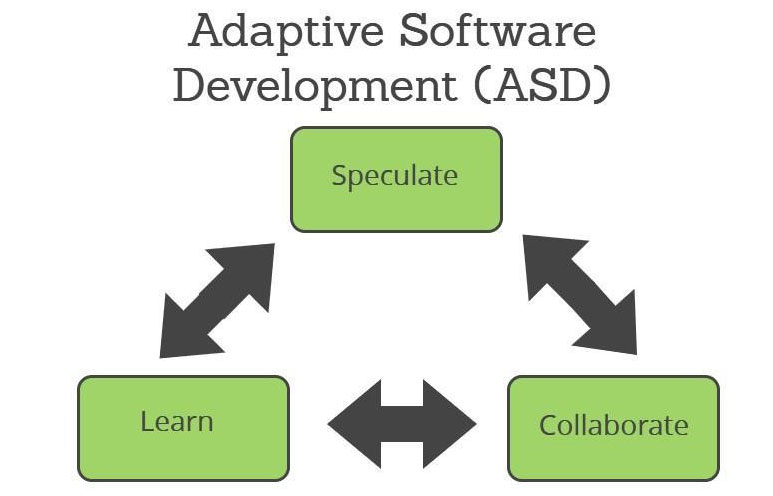
[Extreme Programming](https://en.wikipedia.org/wiki/Extreme_programming) (XP) – or Paired Programming is a methodology developed by [Kent Beck](https://www.kentbeck.com/) in the early 90s. This agile methodology focuses on enhancing interpersonal relationships as a key to success in [software development](https://www.fatbit.com/fab/top-10-custom-software-development-companies/). XP also focuses on promoting teamwork, caring for the learning of developers, and fostering a good working environment. It is characterized by developers working in pairs where one developer programs while the other developer observes; and they switch these roles on a regular basis throughout the Sprint. This way, they enable continuous code review and feedback that enhances code quality and developer capability.



Extreme Programming (XP) promotes continuous feedback between the client and the development teams, fluid communication between all participants, simplicity in the implemented solutions and the readiness to face changes. XP is especially suitable for projects with indistinct and highly changing requirements, and where there is high technical risk.

Adaptive Software Development (ASD)

Adaptive Software Development ([ASD](https://en.wikipedia.org/wiki/Adaptive_software_development)) was developed by [Jim Highsmith](https://www.goodreads.com/author/show/309733.Jim_Highsmith) and Sam Bayer in the early 1990s. It incorporates the principles of continuous adaptation, i.e., *adapt to change and not fight against it*. Adaptive Software Development uses a dynamic development cycle known as Speculate, Collaborate, and Learn. This cycle is dedicated to constant learning and intense collaboration between developers and customers due to the constant change in the business environment.

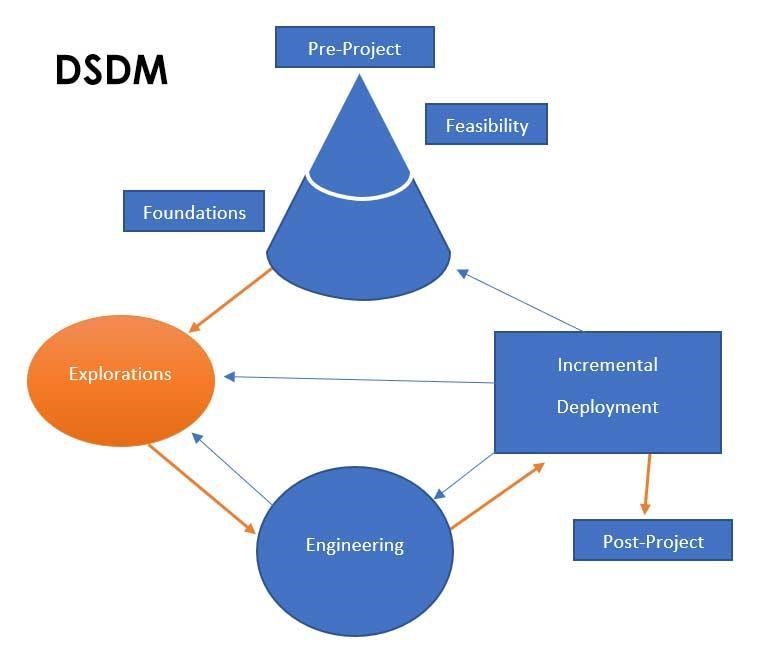


Unlike most Software development methodologies which use a static life cycle i.e., Plan-Design-Build, ASD offers a non-linear iterative life cycle, where each cycle can iterate and be modified while another cycle is being executed.  It points towards Rapid Application Development ([RAD](https://blog.capterra.com/what-is-rapid-application-development/)), which emphasizes development speed to create a high quality, low maintenance product involving the user as much as possible. The main characteristics of ASD are:

1. Speculate: This is the initiation phase of the project where it is necessary to establish the main objectives and goals of the project by understanding the limitations (risk areas) with which the project operates.
2. Collaborate: This is the phase where most of the development is centered, maintaining co-ordination between teams that ensures what is learned by one team is communicated to the rest and does not have to be learned again by other teams from scratch.
3. Learn: The last stage ends with a series of collaboration cycles – the job is to capture what has been learned, both positive and negative. This stage is critical for the effectiveness of the project.

Dynamic Software Development Method (DSDM)

Dynamic Software Development Method ([DSDM](https://www.agilebusiness.org/page/whatisdsdm)) was developed in the year 1994 by a group of vendors and experts in the field of Software development. DSDM focuses on Software projects that are characterized by tight budgets and schedules. It focuses on frequent delivery of product cycles, and development is iterative and incremental.



With Dynamic Software Development Method (DSDM), one can design a roadmap of early and continuous deliveries for the project, implementing an incremental solution, adapting from the feedback obtained throughout the process, and checking that the expected benefits are being met.

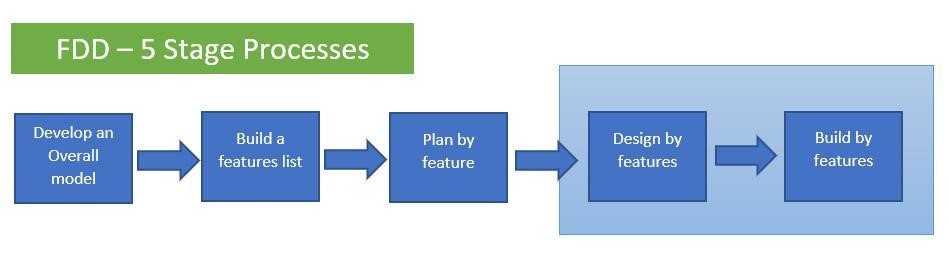
DSDM is an agile model that can undoubtedly help organizations that are used to working on projects to change their mentality and way of working to improve their capacity to deliver value and reduce time to market.

* **Share**
* **Tweet**
* **Share**
* **Share**
* **Pin**
* **Share**

Feature Driven Development (FDD)

Feature Driven Development ([FDD](http://agilemodeling.com/essays/fdd.htm)) methodology is mainly oriented for larger teams with more people than those to whom other agile methodologies such as Scrum are normally applied. FDD was developed by [Jeff De Luca](http://www.jeffdeluca.com/) and Peter Coad in the year 1997. This methodology focuses on short iterations, which allow tangible deliveries of the product in a short period of time (2 weeks).

Projects with multiple teams and a large number of people represent the challenge that not all will be equally talented and disciplined. FDD includes specific activities that help address communication challenges and coordination of such projects.



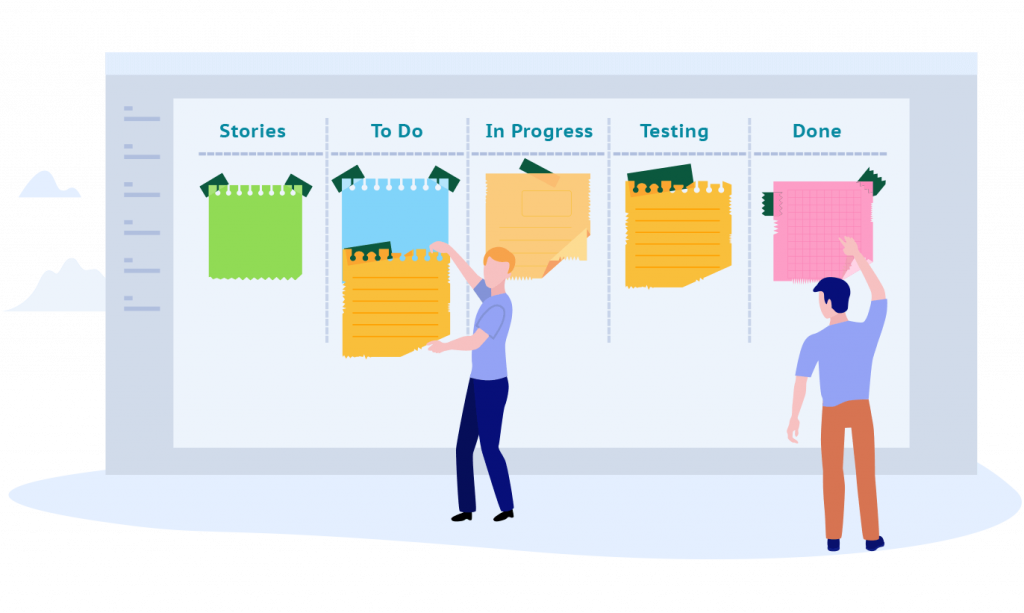
FDD is a 5-stage process, the first 3 of which are sequential and the final two stages are iterative (as shown in the diagram above). All agile methodologies follow a series of principles that make them resemble each other. FDD, however, offers solutions on how to organize the team and how to program the code, which makes it especially viable for large development teams building complex software.

One of the most popular books on the FDD method was published by Stephen Palmer in 2002, titled “[A Practical Guide to Feature-Driven Development](https://books.google.co.in/books/about/A_Practical_Guide_to_Feature_driven_Deve.html?id=NhlFAAAAYAAJ&redir_esc=y)“.

Kanban Method

The Kanban Method was defined by David Anderson in the early -to-mid 2000s, in response to some of the challenges of the various Agile methods, especially Scrum.  These methods, while trying to solve the challenges of traditional/ waterfall methods, became victim to some of the same challenges themselves.

The 2-3 week sprint cycle became too long to wait for many business contexts, the changes required in organizational structure (new roles and responsibilities) and a project management/ planning processes put too much strain on organizations, and many teams found themselves not meeting even sprint-level commitments of scope and quality.  For most organizations, implementing these methods became very disruptive.



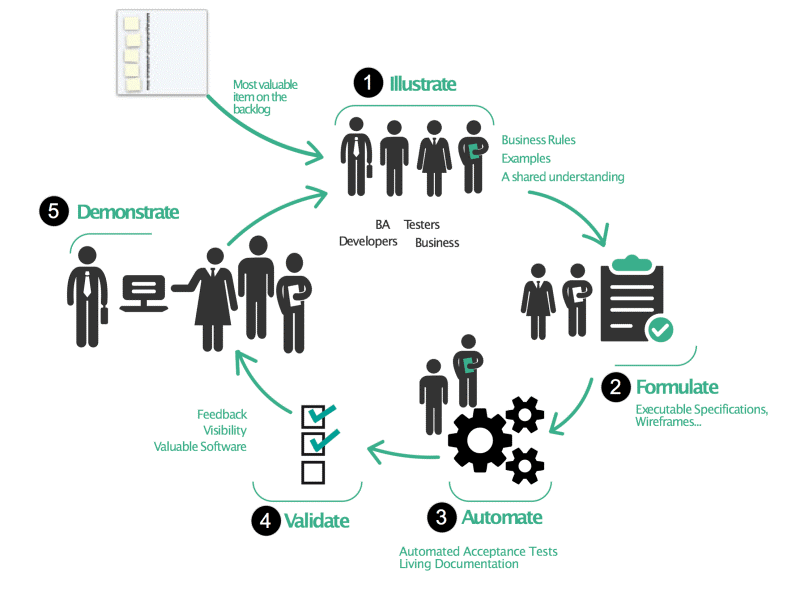
The Kanban Method was defined as the opposite of that – a non-disruptive evolutionary method for improvement, that ultimately enables teams to deliver continuously instead of in time-buckets of 2-3 weeks, get feedback faster and reduce the lead time to deliver value to the customer.

Kanban is a visual system for managing work as it moves through a process. [Kanban](https://www.digite.com/kanban/what-is-kanban/) visualizes both the process (the workflow) and the actual work passing through that process. The goal of Kanban is to identify potential bottlenecks in your process and fix them, so work can flow through it cost-effectively at an optimal speed or throughput.

Kanban is defined as a highly effective and efficient production system. The origin of the Kanban methodology lies in the “just-in-time” (JIT) production processes devised by Toyota, in which cards were used to identify material needs in the production chain.

Behavior Driven Development (BDD)

Behavior Driven Development ([BDD](https://www.agilealliance.org/glossary/bdd/)) is a behavior-oriented agile development methodology. It was created by [Dan North](https://dannorth.net/introducing-bdd/) in 2003 as an evolution of the TDD methodology. Dan North aimed to bring non-technical people together in the process of creating the system’s technical functionality. It happens that when we develop software, we involuntarily fail to include business concepts present in the functionality, resulting in a possible flow for recurring and even serious bugs.



*Source:*[*Johnfergusonsmart.com*](https://johnfergusonsmart.com/behaviour-driven-development-3-minute-rundown/)

BDD uses universal language concepts that encourage collaboration between people with or without technical knowledge in a software project. The BDD development process is based on writing test scenarios and features. These contain the requirements and acceptance criteria for the system behavior. It tells you what the functionality needs to get started, what it will do next, and what the results will be after it is executed.

BDD helps teams more accurately communicate requirements, discover defects early, and build software that remains sustainable over time.

**DevOps**

DevOps is about removing the barriers between traditionally siloed teams, development and operations. Under a DevOps model, development and operations teams work together across the entire software application life cycle, from development and test through deployment to operations.

## **Benefits of DevOps**

* **Speed.** DevOps practices let you move at the velocity you need to innovate faster, adapt to changing markets better, and become more efficient at driving business results.
* **Rapid delivery.**When you increase the pace of releases, you can improve your product faster and build competitive advantage.
* **Reliability.**DevOps practices like [continuous integration and continuous delivery](https://www.synopsys.com/glossary/what-is-cicd.html) can ensure the quality of application updates and infrastructure changes so you can reliably deliver at a more rapid pace while maintaining an optimum experience for end users.
* **Improved collaboration.**Under a DevOps model, developers and operations teams collaborate closely, share responsibilities, and combine their workflows. This reduces inefficiencies and saves time.
* **Security.**You can adopt a DevOps model without sacrificing security by using automated, integrated [security testing tools](https://www.synopsys.com/software-integrity/security-testing.html).

**B. Coding best practices – Industry standards followed to maintain the code quality:**

Importance of code quality and coding standards-

Code quality is crucial for software development. It has a significant impact on the overall quality of software. You can either define a code as good/high-quality code or bad/low-quality code.

Of course, quality, either good or bad, is a subjective matter. Different software development teams may have different definitions depending on the context of coding.

The reason coding standard is so important is highlighted in the real-world stats:

* Five seconds of loading time engages the audience 70% longer than nineteen seconds of loading time.
* 100-millisecond drop in website speed also drops the rate of conversion by almost 7%
* 79% of online shoppers won't return to a website with poor site performance.

## **Why Invest in Good Quality Code?**

Software developers globally adhere to certain coding standards to maintain a quality development environment. Given below are some key benefits when these standards are followed:

* Easier to read
* Easier to maintain
* Easier to understand
* Reusable codes
* Effective performance
* Consistent throughout the solution/software
* Easy knowledge transfer and demonstration

## **Why Code Quality Matters?**

Code quality is the usefulness and maintainability of code throughout the use of the application. It covers good and bad quality codes. It also provides other useful insights about coding standards in software engineering.

Code quality also measures how a code communicates between developers. Even similar software might show a huge difference in coding standards between two developers. However, what matters is consistent code quality throughout the software. A good code is clear and straightforward, bug-free, well tested, documented, refactored, and performant.

Developers spend a sizeable amount of their time addressing technical glitches and fixing bugs. Hence, it is imperative to focus on programming standard, which lies in the code. Code quality isn't a new parameter. It has existed since the 1970s, and it is now a necessity. 90% of companies are using code review tools to improve their code quality.

**Coding Standards Best Practices Help:**

Coding standards best practices are best defined as an assortment of essential rules, best practices, and guidelines to help programmers write good and cleaner code.

Programming standard also promotes sound practices of programming and improves efficiency, while checking that the software is:

* Safe, secure, and hack-proof that can be used without hurting or harming
* Reliable and easy to maintain, which works as per norms, and the codebase also grows
* Testable at the code level
* Compatible or portable with different implementation environments that lead to consistent results.

Both beginner and experienced software developers should follow the coding standards best practices to write good, clean, secure, and reliable code. This reduces future rework as the codebase grows and accelerates marketing time and software performance.

When you follow programming standards, you ensure compliance with the current industry standards, such as IEC and ISO, besides enhanced consistency, security, and code reliability.

## **Benefits and Importance of Code Quality**

The following qualities define coding standards in software engineering:

### 1. Increased Efficiency

Coders spend approximately 75% of their time on debugging. Rather than increasing development costs, executing coding guidelines to identify bugs earlier, and fixing them in due time, is essential. This measure, however, is an excellent long-term strategy for increased efficiency.

Efficiency directly correlates to the speed and performance of software, whereby quality is evaluated. Write reusable codes to reduce resource consumption and use appropriate data types, looping, and function at proper places.

### 2. Reduces Risk of Project Failure

According to this report, 14% of IT projects have been deemed a failure. Implement good quality code to avoid failures and reduce future risks.

### 3. Easy Code Maintenance

Writing a good uniform code makes maintenance easier by decreasing the time of bug detection. If you want to switch IT firms, a new coder could easily navigate the code to identify and fix bugs.

### 4. Creates Clean, Cost-Efficient Code

When you push for coding standards, you get excellent code that can drastically reduce [**software development costs**](https://radixweb.com/blog/cost-to-hire-software-developer) and efforts. Reusing the code saves time, ensuring you can deliver before your deadline.

* **Creation of a software solution using frameworks and libraries to meet specified requirements-**

The goal of frameworks and libraries is to allow developers to focus on creating a unique feature for their web applications without wasting their time. The frameworks and libraries were designed specifically to assist you in improving the performance and efficiency of your web app development processes. They include intriguing features like templates, session management, and database access libraries.

By using the frameworks and libraries ,we feel that it is time saving, reduced efforts, flexibility and highly customizable, secure code, CRUD, Templates system, Suitable for teamwork and provide you with a wide range of web app capabilities, resulting in a much less error-prone program.

Using appropriate framework is essential for a developer because it saves an important time and efforts for building an app. Most of the applications have a common set of functionality such as handling session data validation etc. and web framework prevent a developer from re-writing every time a same code to create a web app.

***Let’s see when you feel need of a web framework:***

**-**When your web app is based on CRUD cases.

**-**When you need a proper separation of the UI and understanding logic but don’t have time to implement a system.

**-**When you find yourself having self-made libraries you use in each of your covering user authentication, session and other usual operations associated with creating a web app

**-**When you are focused to create a CMS in a very short time and you already know the framework

***web frameworks:***

Actually web frameworks are very useful and in many ways they help web developers to build a web app by providing different functions and features. Following are the aspect which shows how frameworks are beneficial:

Saves time:

Biggest advantage of framework is that it reduces time and energy in developing any app because developer doesn’t need to worry about data sanitization, session handling, and error handling and authentication logic. Most of these functions are well taken care by the framework. It avoids head scratching and developer can start writing code for an application straight away without wasting more time with those repetitive coding. It doubles up development process and increases productivity.

Well organized app:

Developer should not have to worry about managing web directories and files. Things get more organized because frameworks already have a good skeleton structure to use. No need to shuffle of files from one place to other. Framework also offers to separate business logic from the interface files.

Flexibility and highly customizable:

If you are a MySQL user and you have been given a PostgreSQL database to use for your app, I’m sure you’d have scratch your head to write the web app from scratch, but the advantage of framework help you to not waste time on studying things that don’t really matters. A few tweaks can help you to ship your application from one platform to another. Add-ons, themes, plugins, widgets are all names for things which develop within framework communities and enable further rapid customization on your application.

Secure code:

Framework makes developer sure that application using good security measurements because framework itself takes care of it. This is another huge advantage of using framework for web development. You as a developer should not have to worry about hacker who can break your app. Framework makes you feel much secure and better.

Say no to re-invent:

Web framework offers many typical components right out of the box such as, user management functionality- which might otherwise take months to build.

Scalable, fast and secure:

Framework are designed to be reused, this leads to quality control on a global scale and so an extremely robust foundation from which to develop your web product from. Such as WordPress; is currently used by over 60 million websites worldwide.

Well supported:

Communities of users and developers spring up around web frameworks where ideas can be shared and knowledge can be captured.

Reduce development time and cost:

For every particular programming language there are web framework created, each has its pro’s and con’s due diligence should be taken when selecting a framework for your web app taking expert advice if necessary.

CRUD:

Frameworks are created to make web development faster and easier. It provides tools to cover the common CRUD cases such as create, read, update, delete. You can find libraries for accessing a database, managing session and cookies, creating templates to display HTML pages and more.

Re-use of code:

Framework also promotes the reuse of code. With a good framework you only need to design for instance, a contact form once. Later you can drop your generic contact from code into all your projects and save yourself some time and efforts.

Templates system:

Most frameworks either provide a templates system or make it easy to add on own template system so that common chunks of HTML that rarely change. For example: header and footer of your page need only to be written once. Inbuilt templates satisfy many developers with design available to create web product quickly.

Easy deployment and maintenance:

Framework based application can be deployed as a web app with windows like functionality. You can support multilingual environment too. With native windows application, user can personalize framework application by rearranging them in many different ways to best fit the way they work. Maintenance is easier because application follow a consistent coding approach, making it easier to understand and maintain the code.

Rapid development Boost Productivity:

Almost all the available frameworks are designed to boost productivity of developer by offering an easy-to-use and easy-to-understand generic application framework. Frameworks also support the rapid prototyping, designing, implementation and deployment of commercially focused application.

Code assistants generate much of the code required and together with an expanding library of components, developers can more rapidly assemble powerful application. Components are also shipped for user management, authority management, server management and common code management.

Suitable for teamwork:

Many frameworks also help you create environment for teamwork. You can let your designers work on the views, database expert work in the models, and let the smart programmer build reusable libraries and plugins etc. also you can let someone build unit tests because they come with tools for that too. For example: PHP frameworks.

***How to select right framework:***

There are many programming languages and so as frameworks available to build a web app but the truth is all frameworks are really just a set of helpful libraries they are building to be leveraged by a particular programming language. When selecting framework for your programming language you will see there are many frameworks available therefore putting strong criteria are necessary as described below:

Requirement list:

Before you start searching for a suitable framework you will need to make a list of requirements about web application and make sure whether a framework is suitable for that purpose

License:

Licenses are important simply because they can have a significant impact on your application. Before you start developing using framework, check out what kind of license the frameworks falls under. While most licenses are pretty liberal to work with and allow you to create commercial application and some of them are not so generous. Find out if license allows you to distribute your application commercially or not.

Popularity:

Choose the framework which is well known, recognized and complete which includes good ideas, the numbers and quality of plugins etc.

Sustainability:

While choosing a framework make sure that it will be able to keep up with you for the duration. This simplifies both the maintenance and upgrading of your application

Techniques:

TO avoid becoming trapped in complexity it is always beneficial to choose an interoperable solution; one that respects best practice in terms of development.

Security:

While choosing a framework to minimize risk make sure it is capable of ensuring security functions such as XSS management.

Documentation:

Well explained, detailed documentation draws in the power users and preacher who then brings on more people and it is the key to its success. With a bad written, confusing document people are going to walk off confused mind and annoyed.

Community support:

Choose the framework which has a friendly community which helps developers new to platform. Communities behind framework can make or break framework. More mannered the communities are more users attracted towards the framework.

Core libraries:

While choosing a framework you as a developer make sure that library you have chosen must be in such as that it frees you from writing repetitive code but still provides a way for you to mess with it if you need more features and controls.

Software pattern:

Almost every framework uses the MVC pattern, which helps you to keep data: the model, the logic, the controller and the user interface, the view, separate from each other. This lets you write a better code which ultimately gives you in better app.

Unit testing:

Frameworks are definitely surplus if lets you write units tests.

Frameworks such as cakePHP, Zend includes code igniter and allows you to create custom tests to check the critical parts of your application.

Bug fixes:

Choose the framework which is not inactive. You don’t want a hacker to tell you that security vulnerability exists in the framework through a page he hacked on your site. You’d rather hear that from the framework developers hopefully with a link to a patch to the issue.

Choose a framework which update often, is open about the bugs it finds and more importantly fixes the bugs as soon as possible.

Ease of installation:

While choosing a right framework for web development ease of installation also plays a very important role.

A framework can be a problem if one has to run through a number of steps just to get it installed and working. This will also bring a problem once the application is ready, tested and needs to be deployed.

Choose a framework which lets you develop and running as rapidly as possible. A framework with ease of installation and deployment adds satisfaction to developer’s life.

Easy extension and availability:

Choose a framework which you can be re- purpose it into a component suitable to reuse in your other application or even better release it to the general public so they can make use of it in their application.

Choose a framework which allows you extended the framework easily with minimal fuss.

While choosing a framework, also remember the availability of plugins. Choose the extension by its quality not by numbers

DB abstraction & ORM:

While choosing a framework, select the one which will allow your web application to become database agnostic. So, you’ll never have to care about the database parts in case you need to switch out database if your framework takes care about it. And the other thing you should be concern about is ORM. ORM allows you to express data as an object and see how it relates to other objects. Such as, Ruby, cakePHP, Django has ORM capabilities.

Hasting requirements:

All the web developers want to build an application on cutting edge platform but it is also depend on client’s budget and demands. So, it may be out of clients given budget to get a dedicated host to place application on you’ll have to make with shared hosting with normal modules and settings.

Learning curve:

When selecting a framework remember to choose one that has the smallest possible learning curve. Some frameworks are flexible when it comes to naming conventions, directory structure and what not’s while others are very strict throwing up errors at tiniest mistakes.

*Choosing a framework must not be taken lightly; it is a long-term commitment. Make sure that you make the right selection!*

* **Specifications - The program functions correctly and meets all of the client’s requirements:**

Specification is a technical document that describes the features and behavior of a software application which we have tried to fix it in this car land motors company website.

Specification helps to get a clear understanding of the product to develop and to minimize software failures.

As per the specifications we have developed step by step by using initialization, analyze, implementation, design, testing and maintenance and have tried to minimize software failures.

**Software Requirements Specification document:**

A Software requirements specification document describes the intended purpose, requirements and nature of a software to be developed. It also includes the yield and cost of the software.

In this document, flight management project is used as an example to explain few points.

## Table of Contents

[](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database/attachment/contents-in-software-requirements-specification-document)

INTRODUCTION

**1.1 PURPOSE**

The purpose of this document is to build an online system to manage flights and passengers to ease the flight management. <<*Include the purpose as applicable to your project*>>

**1.2 DOCUMENT CONVENTIONS**

This document uses the following conventions. <<*Include the conventions as per your application*>>

|  |  |
| --- | --- |
| DB | Database |
| DDB | Distributed Database |
| ER | Entity Relationship |

**1.3 INTENDED AUDIENCE AND READING SUGGESTIONS**

This project is a prototype for the flight management system and it is restricted within the college premises. This has been implemented under the guidance of college professors. This project is useful for the flight management team and as well as to the passengers.

**1.4 PROJECT SCOPE**

The purpose of the online flight management system is to ease flight management and to create a convenient and easy-to-use application for passengers, trying to buy airline tickets. The system is based on a relational database with its flight management and reservation functions. We will have a database server supporting hundreds of major cities around the world as well as thousands of flights by various airline companies. Above all, we hope to provide a comfortable user experience along with the best pricing available.

## 2. OVERALL DESCRIPTION

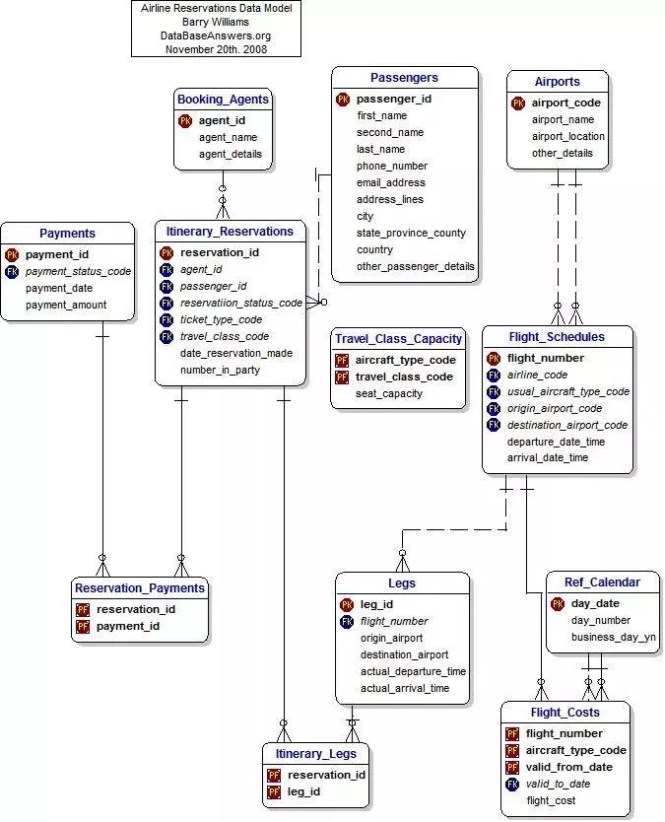
**2.1 PRODUCT PERSPECTIVE**

A distributed airline database system stores the following information.

* **Flight details:**  
  It includes the originating flight terminal and destination terminal, along with the stops in between, the number of seats booked/available seats between two destinations etc.
* **Customer description:**  
  It includes customer code, name, address and phone number. This information may be used for keeping the records of the customer for any emergency or for any other kind of information.
* **Reservation description:**  
  It includes customer details, code number, flight number, date of booking, date of travel.

**2.2 PRODUCT FEATURES**

The major features of airline database system as shown in below [**entity–relationship model**](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model) (**ER model**)

[](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database/attachment/ad-layout-of-airline-database-system)

*The diagram shows the layout of airline database system – entity–relationship model*

**2.3 USER CLASS and CHARACTERISTICS**

Users of the system should be able to retrieve flight information between two given cities with the given date/time of travel from the database. A route from city A to city B is a sequence of connecting flights from A to B such that: a) there are at most two connecting stops, excluding the starting city and destination city of the trip, b) the connecting time is between one to two hours. The system will support two types of user privileges, Customer, and Employee. Customers will have access to customer functions, and the employees will have access to both customer and flight management functions.

Each flight has a limited number of available seats. There are a number of flights which depart from or arrive at different cities on different dates and time.

**2.4 OPERATING ENVIRONMENT**

Operating environment for the airline management system is as listed below.  <<*Include the details as per your application*>>

* distributed database
* client/server system
* Operating system: Windows.
* database: sql+ database
* platform: vb.net/Java/PHP

**2.5 DESIGN and IMPLEMENTATION CONSTRAINTS**

1. The global schema, fragmentation schema, and allocation schema.
2. SQL commands for above queries/applications
3. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.
4. Implement the database at least using a centralized database management system.

**2.6 ASSUMPTION DEPENDENCIES**

Let us assume that this is a distributed airline management system and it is used in the following application:

* A request for booking/cancellation of a flight from any source to any destination, giving connected flights in case no direct flight between the specified Source-Destination pair exist.
* Calculation of high fliers (most frequent fliers) and calculating appropriate reward points for these fliers.

Assuming both the transactions are single transactions, we have designed a distributed database that is geographically dispersed at four cities Delhi, Mumbai, Chennai, and Kolkatta as shown in fig. below.

## 3. SYSTEM FEATURES

* **DESCRIPTION and PRIORITY**

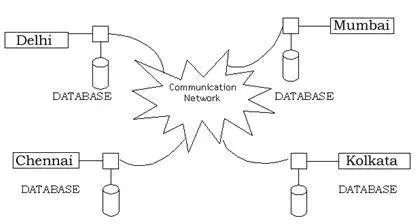
The airline reservation system maintains information on flights, classes of seats, personal preferences, prices, and bookings. Of course, this project has a high priority because it is very difficult to travel across countries without prior reservations.

* **STIMULUS/RESPONSE SEQUENCES**
  + Search for Airline Flights for two Travel cities
  + Displays a detailed list of available flights and make a “Reservation” or Book a ticket on a particular flight.
  + Cancel an existing Reservation.
* **FUNCTIONAL REQUIREMENTS**

Other system features include:

**DISTRIBUTED DATABASE:**

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as shown in below figure.

[](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database/attachment/ad-distributed-database-located-at-four-different-cities)

*Distributed database located in four different cities*

**CLIENT/SERVER SYSTEM**

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

* Some sites are client sites and others are server sites.
* All the data resides at the server sites.
* All applications execute at the client sites.

## 4. EXTERNAL INTERFACE REQUIREMENTS

**4.1 USER INTERFACES**

* Front-end software: Vb.net version
* Back-end software: SQL+

**4.2 HARDWARE INTERFACES**

* Windows.
* A browser which supports CGI, HTML & Javascript.

**4.3 SOFTWARE INTERFACES**

Following are the software used for the flight management online application. <<*Include the software details as per your project*>>

|  |  |
| --- | --- |
| **Software used** | **Description** |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save the flight records, passengers records we have chosen SQL+ database. |
| VB.Net | To implement the project we have chosen Vb.Net language for its more interactive support. |

**4.4 COMMUNICATION INTERFACES**

This project supports all types of web browsers. We are using simple electronic forms for the reservation forms, ticket booking etc.

## 5. NONFUNCTIONAL REQUIREMENTS

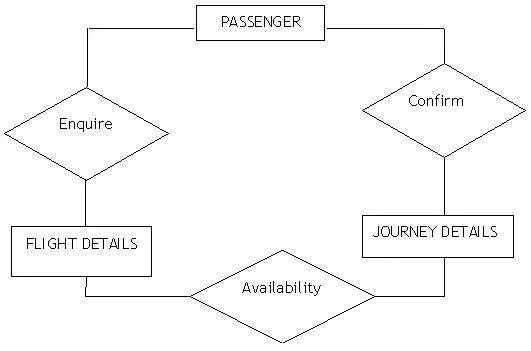
**5.1 PERFORMANCE REQUIREMENTS**

The steps involved to perform the implementation of airline database are as listed below.

**A) E-R DIAGRAM**

The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

* **ENTITIES:**Which specify distinct real-world items in an application.
* **PROPERTIES/ATTRIBUTES:** Which specify properties of an entity and relationships.
* **RELATIONSHIPS:** Which connect entities and represent meaningful dependencies between them.

[](https://krazytech.com/projects/sample-software-requirements-specificationsrs-report-airline-database/attachment/ad-er-diagram-of-airline-database)

*the diagram shows the ER diagram of airline database*

**B) NORMALIZATION:**

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

**5.2 SAFETY REQUIREMENTS**

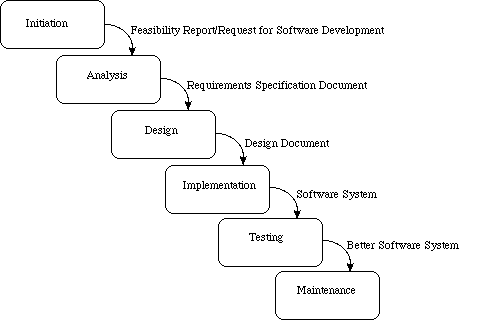
If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

**5.3 SECURITY REQUIREMENTS**

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

**5.4 SOFTWARE QUALITY ATTRIBUTES**

* **AVAILABILITY:** The flight should be available on the specified date and specified time as many customers are doing advance reservations.
* **CORRECTNESS:** The flight should reach start from correct start terminal and should reach the correct destination.
* **MAINTAINABILITY:** The administrators and flight in chargers should maintain correct schedules of flights.
* **USABILITY:** The flight schedules should satisfy a maximum number of customers needs.

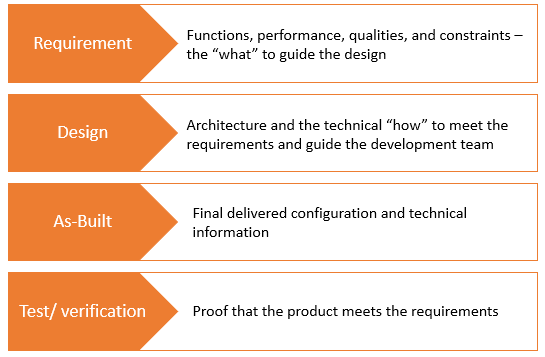


We have tried to describe system behavior under specific conditions and include the product features and functions which web & app developers must add to the solution.

**Functional Requirements of a website:**

* Business Rules.
* Transaction corrections, adjustments and cancellations.
* Administrative functions.
* Authentication.
* Authorization levels.
* Audit Tracking.
* External Interfaces.
* Certification Requirements.

Requirements and specifications are very important components in the development of any embedded system. Requirements analysis is the first step in the system design process, where a user's requirements should be clarified and documented to generate the corresponding specifications.



**References:**

* [**https://www.nngroup.com/articles/principles-visual-design**](https://www.nngroup.com/articles/principles-visual-design)**( By Kelley Gorden )**
* Lupton, E. (2008). Graphic Design: The New Basics. New York: Princeton Architectural Press.
* Poulin, R. (2018). The Language of Graphic Design. Beverly: Quarto Publishing Group USA, Inc.
* https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board
* <https://radixweb.com/blog/cost-to-hire-software-developer>
* https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html
* Lucid chart app
* Excel Sheet
* Fundamentals of database systems by ramez elmarsi and shamkant b.navathe