**WEEK:01**

**1.DISCUSS SUCCESS AND FAILURE STORIES**

# Successful story of software

1. **FAMPAY:**



**FamPay** is India's first neo bank designed specifically for teenagers.

Founded in 2019 by Sambhav Jain and Kush Taneja, it allows minors to make UPI, and card payments without needing a bank account.

A company offering a family-oriented payment application that enables teenagers to receive money from their parents and spend it digitally without opening a bank account.

FamPay is a fintech company that offers a payments app for teenagers and their families. Founded in 2019 by IIT Roorkee graduates Kush Taneja and Sambhav Jain, FamPay aims to drive 250M+ teenagers on a journey from cash to digital.

**Key Features:**

**FamX Card:** A numberless card designed for secure payments, allowing users to make tap and pay transactions.

**Custom UPI ID:** A personalized UPI ID for making and receiving payments, with customizable QR themes.

**FamCoins and Cashbacks:** A rewards program offering cashbacks and FamCoins for transactions, redeemable for discounts and offers.

**Design Your Own Card**: A service allowing users to customize their card design with personal doodles or themes.

**Security Measures:**

**Partnership with RBI-regulated entity:** FamPay has partnered with an RBI-regulated entity to ensure secure payments.

**No linking of bank account:** FamPay accounts are reloadable and do

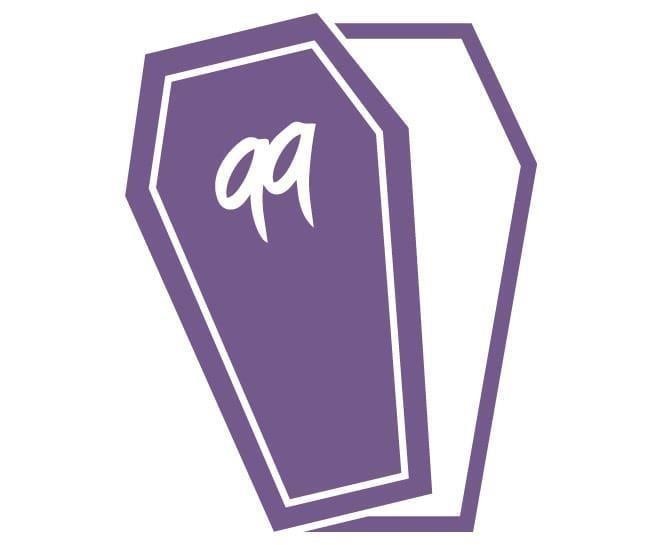
**2)WHERE IS MY TRAIN:**



* **Where Is My Train** is an [Android](https://en.wikipedia.org/wiki/Android_(operating_system)) application owned by [Google](https://en.wikipedia.org/wiki/Google) for tracking the live status of trains run by [Indian Railways.](https://en.wikipedia.org/wiki/Indian_Railways) The application was created by Sigmoid Labs, a team of former [TiVo Corporation](https://en.wikipedia.org/wiki/TiVo_Corporation) developers. The company was acquired by Google in
* Get live train status of Indian Railways anytime, anywhere. When you are traveling on a train, this feature can work without internet or GPS as it uses cell tower information to find the location. You can share current train location with your friends & family via the share feature.
* You can also set an alarm to wake you up at a fixed time before your railway station arrives
* Is an Android application
* Founder of this application Ahmed Nizam Mohaideen.
* This was started on 20 June 2015.
* Is a unique train app that displays live train status and up-to-date schedules.
* The app can function offline without needing internet or GPS.
* This app was invested up to 30-40 million.
* In the year of 2018 google acquired where is my train.
* In 2018 the app reversed a nomination for India’s best apps of 2018 in the play store.
* This apps includes 8 language English , kannada , tamil etc
* The app is free to download and use, ensuring that its benefits reach a large audience without financial barriers.
* By providing accurate train schedules and reducing the need for inquiries, the app saves passengers both time and effort.
* Travelers no longer have to rely solely on station announcements or station staff for updates
* It helps passengers locate their train's platform number.

# Failure story

**A). 99 DRESSES:**



As per research, out of 10.000 funded start ups, only a single one becomes a unicorn. Every year a number of start ups fail, even those with VC backing and support. But, with every one of these failures, we learn something.

* 99 dresses was a fashion application.
* This app was Founded by Nikki Durkin.
* It was started on 2010 in Australia. Total funding amount $105.7k  There are 4 investors.
* The platform launched an IOS app
* Low quality dresses
* 99 dresses was established by the 18 old Nikki who had no idea about the tech business.
* It successfully operated for 1 year.
* The key issues was that the organization did not raise enough income to maintain its task has the organization plan of action dependent on the exchange cost
* 99 dresses failed due to a loss of investors trust and lakh of fund.

A women's fashion startup that's so compelling even tech dudes get it? That may sound like a girl's fantasy, but it's become reality for the 20-year old Australian entrepreneur Nikki Durkin, founder of[99dresses.com](http://www.99dresses.com/shop) who [created a buzz yesterday at the YCombinator Winter 2012 Demo Day](http://finance.yahoo.com/news/99dresses-emerges-hottest-startup-ycombinator-114900267.html?utm_source=twitterfeed&utm_medium=twitter). 500Startups George Kellerman and former Mashable editor [Ben Parr](http://blogs.forbes.com/velocity/) gave 99Dresses high marks among the 66 ventures that participated, according to the company.

It's not hard to see why. First, the concept is exceedingly simple: 99Dresses is a platform for swapping frocks using virtual currency. Durkin told me, "Upload your quality unwanted clothes, shoes and accessories into the Infinite Closet, sell them to other girls for a virtual currency called ‘buttons’ then spend those buttons on anything you want in the Infinite Closet. You might buy an outfit, rock it at a party then resell it in a click for another woman to enjoy and go buy something new."

Dresses can be found in a variety of sizes, hail from a diverse portfolio of designers and can be had for between 6 and 100+ buttons. Says Durkin: "99dresses is creating a new model for fashion consumption for young women that is ridiculously addictive. 99dresses is crack for women. They love it and they keep coming back for more. We’ve seen this happen in Australia and we are confident that this will also happen in the US."

**B) HOTELSAROUNDYOU:**



A Mumbai-based, hotel budget startup, HotelsAroundYou was also known as Instant Hotels Around You Pvt Ltd facilitates users with the limited location-based hotel’s feature and also assures the day’s best price deal for a hotel. It was a last minute hotel booking platform making the users experience easy with secure hotel booking at the last minute by booking the same-day unsold hotel rooms easily at the best price.

|  |
| --- |
| The seed funding they had raised from the market all goes in vain. Other than this, it was a setback to those investors who invested in their company and for their users. Major competitors for the HotelsAroundYou was Rooms Tonite, Night stay, Stayzilla, [Oyo Rooms,](https://www.craftdrivenresearch.com/hospitalityindustry/) Airbnb, MakeMyTrip, Goibibo and various other alternate online platforms for room booking start ups too. They had wasted so much time on a service, which is already providing by most of the experienced players of the hospitality industry.  As we always discuss the loss of employment cause it is linked in many ways other than sitting job done by the workers, the loss of employment. Due to the shutting down of the  HotelsAroundYou, employees left their jobs and the biggest loss to the country’s economic growth. |

* Hotels Around You was an India focused service centered on last minute.
* Hotels Around You is a short stay bookings.

It offered users to book hotels for two types of stays

1.Transit option

2. Night use hotel

* Name of founder Animesh Chudhary, Harsha nallur, Moshin Dingankar.
* Launched in 2013 in Mumbai
* Number of employees are 10
* Total funding amount 125.4k and number of investors is 2
* It was a last night hotel booking platform making the user experience easy with s hotel booking at the last minute by booking the same day unsold hotel rooms easily at the best prize.
* Due to shutting down of the hotels around you , employees left their jobs and the biggest loss to the countries economic growth .

**03).Enact the importance of ethical practices**

**Ethical Practice Standards**

* Each standard progresses through four levels of impact:
* Foundation Level
* At this level you will:
* Take responsibility for your actions
* Act consistently with relevant regulations and law
* Handle personal data and information in a professional manner
* Demonstrate honesty in dealings with others

**Associate Level**

* At this level you will:
* Make responsibility choices about your work, apply principles and values.
* Consider the purpose and implication of actions, decisions and people practices for all stakeholders.
* Provide explanations and reasons for the choices you make and the advice you provide.
* Demonstrate professionalism and consistency in what you say and do in order to build trust.

**Chartered Member Level**

* level you will:
* Make responsible decisions by considering different ethical perspectives, and finding the best possible way forward for all stakeholders.
* Coach and Influence managers and leaders to consider the implication of their decisions on stakeholders.
* Challenge decisions and actions which are not ethical, explaining the organization risks.
* Encourage transparency in decision-making and communication where possible.

**Chartered Fellow Level**

* At this level you will:
* Make responsible decisions by balancing different ethical perspectives, and shape how ethics inform wider decision-making and governance.
* Coach and influence senior leaders to consider the ethical impact of their decisions in the short long-term
* Take a visible lead in solving ethical dilemmas, considering how they will play out beyond the organization.
* Surface the unsaid in leadership discussions to enable transparency and improved decision making.

**Example of Ethics in software engineering:**

1. PUBLIC

2. PRODUCT

3. JUDGMENT

4. MANAGEMENT

5. Client and Employer

6. Profession

7. Colleagues

8. Self

**PUBLIC:**

* Software engineers shall act in a manner that is in the best interests of their client and employer,

consistent with the public interest. In particular, software engineers shall, as appropriate:

* Accept full responsibility for their own work.
* Moderate the interests of the software engineer, the employer, the client and the users with the public

good.

* Be fair and avoid deception in all statements, particularly public ones, concerning software

**PRODUCT**

* Software engineers shall ensure that their products and related modifications meet the highest

professional standards possible. In particular, software engineers shall, as appropriate:

* Strive for high quality, acceptable cost and a reasonable schedule, ensuring significant trade-offs are

clear to and accepted by the employer and the client, and are available for consideration by the user

and the public.

* Ensure proper and achievable goals and objectives for any project on which they work or propose.
* Ensure that they are qualified for any project on which they work or propose to work by an

Appropriate combination of education and training, and experience.

* Work to follow professional standards, when available, that are most appropriate for the task at

hand, departing from these only when ethically or technically justified

**JUDGMENT**

* Software engineers shall maintain integrity and independence in their professional judgment. In

particular, software engineers shall, as appropriate:

* Only endorse documents either prepared under their supervision or within their areas of competence

and with which they are in agreement.

* Temper all technical judgments by the need to support and maintain human values.
* Maintain professional objectivity with respect to any software or related documents they are asked to

evaluate.

* Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or

escaped.

**MANAGEMENT**

* Software engineering managers and leaders shall subscribe to and promote an ethical approach to the

management of software development and maintenance. In particular, those managing or leading

software engineers shall, as appropriate:

* Ensure good management for any project on which they work, including Effective procedures for

promotion of quality and reduction of risk.

* Ensure that software engineers are informed of standards before being held to them.
* Ensure that software engineers know the employer’s policies and procedures for protecting

passwords, files and information that is confidential to the employer or confidential to others

**Client and Employer**

* Software engineers shall act in a manner that isin the best interests oftheir client and employer,

Consistent with the public interest.

* In particular, software engineers shall, as appropriate: Provide service in their areas ocompetence,

being honest and forthright about any limitations of their experience and education.

* Not knowingly use software that is obtained or retained either illegally or unethically.
* Use the property of a client or employer only in ways properly authorized, and with the client’s

Or employer’s knowledge and consent.

* Ensure that any document upon which they rely has been approved, when required, by someone

Authorized to approve it.

**Profession**

* Software engineers shall advance the integrity and reputation of the profession consistent with the

public interest.

* In particular, software engineers shall, as appropriate:
* Help develop an organizational environment favorable to acting ethically.
* Promote public knowledge of software engineering.
* Extend software engineering knowledge by appropriate participation in professional

organizations,meetings and publications.

* Support, as members of a profession, other software engineers striving to follow this Code.

**Colleagues**

* Software engineers shall be fair to and supportive of their colleagues. In particular, software

engineers shall, as appropriate.

* Encourage colleagues to adhere to this Code.
* Assist colleagues in professional development
* Credit fully the work of others and refrain from taking undue credit.
* Review the work of others in an objective, candid, and properly- documented way.
* Give a fair hearing to the opinions, concerns, or complaints of a colleague.
* Assist colleagues in being fully aware of current standard work practices including policies and

procedures for protecting passwords, files and other confidential information, and security measures

in general.

**Self**

* Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession. In particular, software engineers shall continually endeavor to.
* Further their knowledge of developments in the analysis, specification, design, development,

maintenance and testing of software and related documents, together with the management of the

development process.

* Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time.
* Improve their ability to produce accurate, informative, and well-written documentation

**WEEK-02**

**Overview SDLC**

The Software Development Life Cycle (SDLC) refers to a methodology with clearly defined

processes for creating high-quality software. in detail, the SDLC methodology focuses on the

following phases of software development:

* Requirement
* analysis
* Planning
* Software design such as architectural design
* Software development
* Testing
* Deployment

This article will explain how SDLC works, dive deeper in each of the phases, and provide you with

examples to get a better understanding of each phase.

**What is the software development life cycle?**

SDLC or the Software Development Life Cycle is a process that produces software with the highest

quality and lowest cost in the shortest time possible. SDLC provides a well-structured flow of phases

that help an organization to quickly produce high-quality software which is well-tested and ready for

production use.

The SDLC involves six phases as explained in the introduction. Popular SDLC models include the

waterfall model, spiral model, and agile model.

So, how does the Software Development Life Cycle work?

**How the SDLC Works**

SDLC works by lowering the cost of software development while simultaneously improving quality and

shortening production time. SDLC achieves these apparently divergent goals by following a plan that

removes the typical pitfalls of software development projects. That plan starts by evaluating existing

systems for deficiencies.

Next, it defines the requirements of the new system. It then creates the software through the stages of

analysis, planning, design, development, testing, and deployment. By anticipating costly mistakes like

failing to ask the end-user or client for feedback, SLDC can eliminate redundant rework and after-the fact fixes.

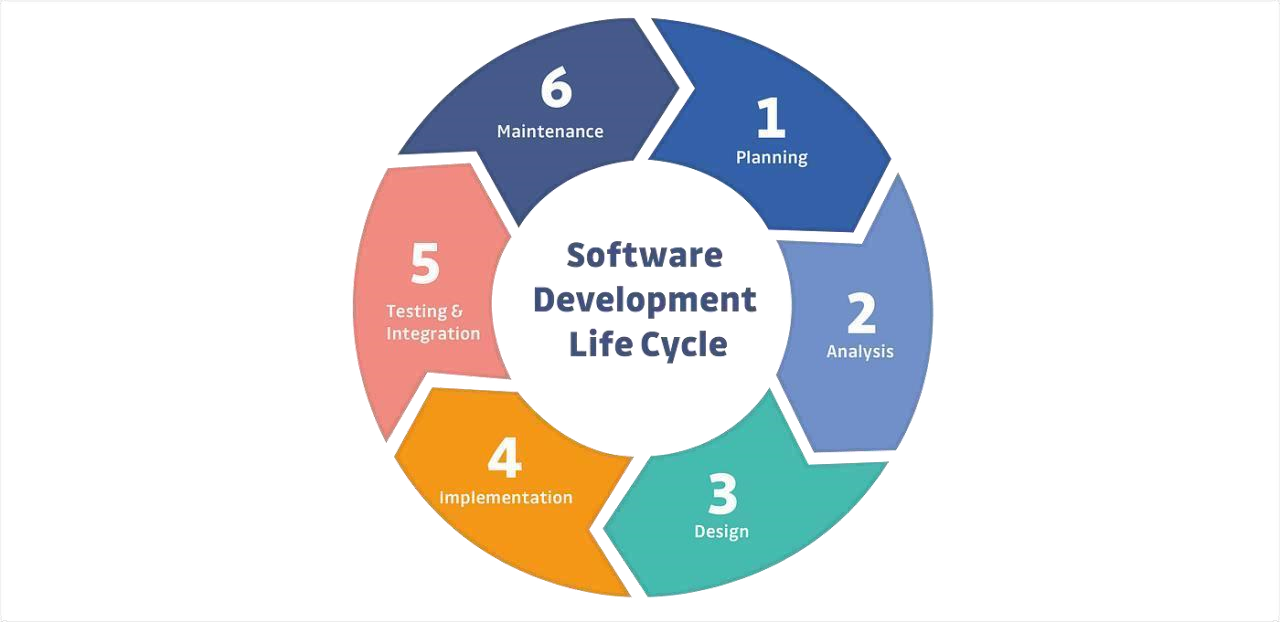
It’s also important to know that there is a strong focus on the testing phase. As the SDLC is a

repetitive methodology, you have to ensure code quality at every cycle. Many organizations tend to

spend few efforts on testing while a stronger focus on testing can save them a lot of rework, time, and

money. Be smart and write the right types of tests.

Next, let’s explore the different stages of the Software Development Life Cycle.



**Stages and Best Practices**

Following the best practices and/or stages of SDLC ensures the process works in a smooth, efficient,

and productive way.

**1.Identify the Current Problems**

“What are the current problems?” This stage of the SDLC means getting input from all stakeholders,

including customers, salespeople, industry experts, and programmers. Learn the strengths and

weaknesses of the current system with improvement as the goal.

**2.Plan**

“What do we want?” In this stage of the SDLC, the team determines the cost and resources required for

implementing the analyzed requirements. It also details the risks involved and provides sub-plans for

softening those risks.

In other words, the team should determine the feasibility of the project and how they can implement the

project successfully with the lowest risk in mind.

**3.Design**

“How will we get what we want?” This phase of the SDLC starts by turning the software specifications

into a design plan called the Design Specification. All stakeholders then review this plan and offer

feedback and suggestions. It’s crucial to have a plan for collecting and incorporating stakeholder input

into this document. Failure at this stage will almost certainly result in cost overruns at best and the total

collapse of the project at worst

**4.Build**

“Let’s create what we want.”

At this stage, the actual development starts. It’s important that every developer sticks to the agreed

blueprint. Also, make sure you have proper guidelines in place about the code style and practices.

For example, define a nomenclature for files or define a variable naming style such as camel Case. This

will help your team to produce organized and consistent code that is easier to understand but also to test

during the next phase.

**5. Code Test**

“Did we get what we want?” In this stage, we test for defects and deficiencies. We fix those issues until

the product meets the original specifications.

In short, we want to verify if the code meets the defined requirements.

**6. Software Deployment**

“Let’s start using what we got.”

At this stage, the goal is to deploy the software to the production environment so users can start using

the product. However, many organizations choose to move the product through different deployment

environments such as a testing or staging environment.

This allows any stakeholders to safely play with the product before releasing it to the market. Besides,

this allows any final mistakes to be caught before releasing the product.

Extra: Software Maintenance

“Let’s get this closer to what we want.” The plan almost never turns out perfect when it meets reality.

Further, as conditions in the real world change, we need to update and advance the software to match.

The Dev Ops movement has changed the SDLC in some ways. Developers are now responsible for more

and more steps of the entire development process. We also see the value of shifting left. When

development and Ops teams use the same toolset to track performance and pin down defects from

inception to the retirement of an application, this provides a common language and faster handoffs

between teams.

Application performance monitoring (APM) tools can be used in a development, QA, and production

environment. This keeps everyone using the same toolset across the entire development lifecycle.

Examples: The most common SDLC examples or SDLC models are listed below.

**Waterfall Model**

This SDLC model is the oldest and most straightforward. With this methodology, we finish one phase

and then start the next. Each phase has its own mini-plan and each phase “waterfalls” into the next. The

biggest drawback of this model is that small details left incomplete can hold up the entire process.

**Agile Model**

The Agile SDLC model separates the product into cycles and delivers a working product very quickly.

This methodology produces a succession of releases. Testing of each release feeds back info that’s

incorporated into the next version. According to Robert Half, the drawback of this model is that the

heavy emphasis on customer interaction can lead the project in the wrong direction in some cases.

**Iterative Model**

This SDLC model emphasizes repetition. Developers create a version very quickly and for relatively

little cost, then test and improve it through rapid and successive versions. One big disadvantage here is

that it can eat up resources fast if left unchecked.

**V-Shaped Model**

An extension of the waterfall model, this SDLC methodology tests at each stage of development. As

with waterfall, this process can run into roadblocks.

**Big Bang Model**

This high-risk SDLC model throws most of its resources at development and works best for

small projects. It lacks the thorough requirements definition stage of the other methods.

**Spiral Model**

The most flexible of the SDLC models, the spiral model is similar to the iterative model in its emphasis

on repetition. The spiral model goes through the planning, design, build and test phases over and over,

with gradual improvements at each pass.

**1.Case Study to Understand SDLC On Mobile Development Application**

**Life cycle.**

**Introduction**

Mobile apps are designed to run on specific mobile operating systems such as ios,Android and Windows Phone. When a mobile app is downloaded and installed on a device, it is stored in the device's memory and is launched using the device's operating system.

When a user opens a mobile app, the app communicates with the device's [operating system](https://www.techtarget.com/whatis/definition/operating-system-OS) and other built-in software components to access the device's hardware and services such as the camera, [GPS](https://www.techtarget.com/searchmobilecomputing/definition/Global-Positioning-System) and internet connection. The app then uses this information to provide its specific functions and services to the user.



### 1. The Research/Planning Stage

Your first step should be to dive deep into the research phase- figure out your market and existing competitor apps. Brainstorm on details like the purpose of your app, your target audience, preferred platforms, app development language & frameworks, features your competitor app offers (and if you’d want the same/different ones), a timeline of development & launch, and how you’d want to market it.

**2. Wireframes/Analysis**

The next step is where you document and wireframe your application. Drawing detailed sketches of your vision of the app (how it would look, the features it would have, etc) greatly helps in bringing it to life in the later stages. Post sketching, wireframes come to refine all your ideas. Now you can arrange all your design components accurately and see if there are any visible usability issues

**3. Design**

Once prototyping is done, you can dive into coding. Here, your UX and UI designers take over. Your user experience (UX) designer builds interaction between different design elements while your user interface (UI) designer builds the overall look and feel of your app. What you get here are visual directions & blueprints.

## 4.Implementation Lifecycle: After Launch

The implementation lifecycle of a mobile app begins once the app is launched and available to users. It involves on going activities aimed at maximizing the app's performance, user satisfaction, and overall success. Let's explore the key steps after the launch of your mobile application.

### 5. Testing

In the mobile app development lifecycle, testing plays a very important role. It’s often a good idea to start testing early as this could help with keeping the final costs in check. At this stage, the app should ideally be tested for every aspect including usability, compatibility, interface, security checks, stress, and performance. Inviting some of your target audience to test it would be a good idea

**6. Maintenance & Updates**

To keep the app running smoothly and address any issues or bugs that may arise, regular maintenance and updates are necessary. This phase involves monitoring the app's performance, collecting user feedback, and addressing reported issues promptly. Regular updates are released to fix bugs, improve performance, and introduce new features based on user needs and market trends.

**Challenges Faced**

## Unique Needs of Mobile App Projects

Many CI/CD tools are built for general purposes, requiring substantial customization to suit mobile app projects. However, App circle is designed specifically for mobile CI/CD, addressing the specific needs of mobile app development without additional integrations or tools. Learn more about [Appcircle’s mobile-first approach](https://appcircle.io/platforms" \t "_blank).

## 2. Inefficiency and Errors in Manual Operations

Without a mobile CI/CD flow, manual handoffs in the mobile app lifecycle are inefficient and prone to error. Appcircle automates the entire mobile CI/CD process, reducing human intervention and improving productivity while maintaining quality. Discover the power of [end-to-end automation](https://appcircle.io/build).

## 3. Complex iOS and Android CI/CD Requirements

Establishing separate CI/CD flows for iOS and Android can be challenging due to differing requirements. App circle simplifies this with an easy-to-start, low-maintenance platform where you can connect your repository and configure builds, allowing App circle to handle the complexity of mobile OS requirements.

## 4. Balancing Simplicity and Flexibility

Traditional CI/CD tools often force a choice between flexibility and simplicity. Appcircle offers both, allowing you to start quickly with an intuitive interface and expand or customize workflows as needed. Learn more about [App circle’s flexible workflows](https://appcircle.io/integrations) to suit any mobile CI/CD needs.

**Key Features of the Platform**

**1. User-Centric**

**2. Modern Technology**

**3. Cross-platform App**

**4. Application Security**

**5. Offline Functionality**

**6. Voice Search**

**7. Location Tracking**

**8. Social Integration**

**9. Easy Payment Gateway**

**10. App Analytics**

**11. Regular Updates**

## Conclusion

For complex iOS and Android projects, mobile CI/CD is essential. App circle makes it easy to set up a complete mobile CI/CD pipeline, providing full coverage from development to distribution with automation, flexibility, and robust team management. Start your journey with App circle and optimize your mobile app lifecycle to achieve the best results.

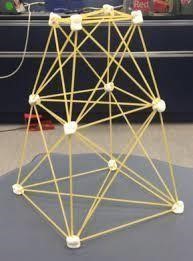
**02). Organize and play games to understand the agile process like, Morning wake**

**up game.**

**A)The Marshmallow Challenges**:

The marshmallow challenge was introduced by Tom Wujec. The Purpose of the challenge is to build the

Team Coordination, team Work, Patience, Communication and Time Management.



**Objectives:**

The goal is to build a freestanding structure using limited materials that can hold a marshmallow without

it falling.

**Requirement Needed for this Challenge Was:**

a. 20 sticks of spaghetti

b. 8 Marsh Mallow

* Build the tallest Free-standing Structure in just 18 minutes using no more than 20 sticks of Spaghetti.

one yard of String, and 08 Marsh Mallow.

* The structure has to stand firmly on its own, it cannot be propped upTeams cannot hold on to the

Structure when the time runs out. Those touching or supporting the Structure at the end of the exercise

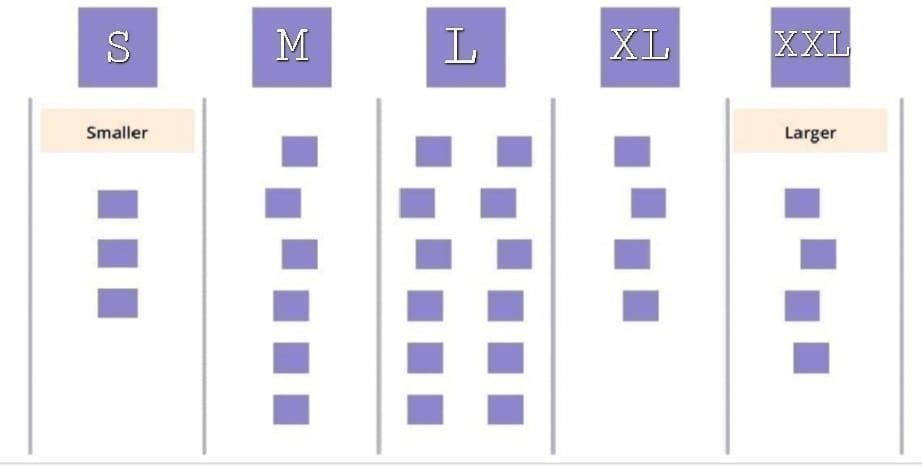
will be disqualified.

* The Marshmallow Challenge is a fun and engaging way to build teamwork and problem-solving skills.

It's an excellent exercise for corporate teams, students, and anyone looking to boost their collaboration

and creativity skills.

**B) White Elephant Sizing**



**Overview:**

This game is mainly meant for the analyze of the value from user interface according to the mindset of

the team members.

* Group the user stories according to their relative size/efforts taken to build a software. Ensure that each

teammates gets a chance.

* Learn how user stories are been captured.
* Actively participate in a fun way.
* White Elephant Sizing game mainly is to get the relative size of an agile project and the size of the

individual stories before the project starts.

* It mainly gives opportunity to everyone in the team to raise their voice and contribute equally.

**Game:**

The game is based on food application named Zomato (only an example).

**Rules:**

If another person in the hall feels wrong can replace the card placed on the board to different column (by

giving a solid reason)

**Materials:**

* Blackboard
* A set of prepared user interface stories.
* Sticky note pads for attaching the related size/value

**User Interfaces:**

* UI 1: Logo interface
* UI 2: Login /Signup Screen
* UI 3: Account details
* UI 4: Menu Screen: Displays all food options.
* UI 5: Customization Screen
* UI 6: Cart Screen
* UI 7: Checkout Screen
* UI 8: Real-time updates on the order status/Location
* UI 9: Settings Screen
* UI 10: Offers
* UI 11: History

**Procedure:**

* Create five columns on the board and divide it as S, M, L, XL and XXL
* The set of prepared user stories on notes. It will be summary with brief description of the different user

stories on different notes (tape or sticky notes).

* After the multiple user screens are explained in game. After reading the person only has to assign the

card to a column. By giving the explanation.

* It will be continued until the user story cards come to an end.

**C).Easter Egg Challenge**

A challenge, inspired by this article, that on first sight looks fun and irrelevant, represents in many ways

how teams operate, and how that is impacting their productivity. While painting eggs, cutting them out,

checking the quality of the delivered product seems extremely easy, the reality shows that even

supposedly easy tasks can become difficult and hectic when the participant’s mindset is not right.



**Requirement needed for this challenge are:**

1. A bunch of papers with unpainted Easter eggs on them

2. Several boxes of Color pencils and some scissors

**Objectives:**

The challenge is simple: During the challenge the teams will be instructed to deliver the painted eggs

according to the pre-defined requirements, on time.

**Procedure:**

* Groups of 4 or 5individuals per team.
* Color and design the 12 unpainted ester eggs.
* After coloring the Easter eggs take the scissor and cut in the shape of eggs.
* This activity should be finished by 15 minutes.
* Then which team has finished the challenge in given time with well-designed eggs will be the winner.

**03). Create Jira (similar tool) Account and learn Interface**

**Jira Software:**

**JIRA** is a software development tool used for project management and issue tracking. It is a popular tool

among software development teams to plan, track, and release software projects. JIRA provides a

centralized platform for managing tasks, bugs, and other types of issues and it helps teams to organize and

prioritize their work. The tool integrates with other software development tools and has a variety of

customizable features and workflows that allow teams to adapt it to their specific needs. Additionally,

JIRA also provides various reporting and dashboard features that help teams to stay on top of their work

and make Data-driven decisions.

**Jira Features:**

* JQL (Jira Query Language).
* Creates advanced Dashboards.
* Add-ons allows customized features, unique, design
* Reports.

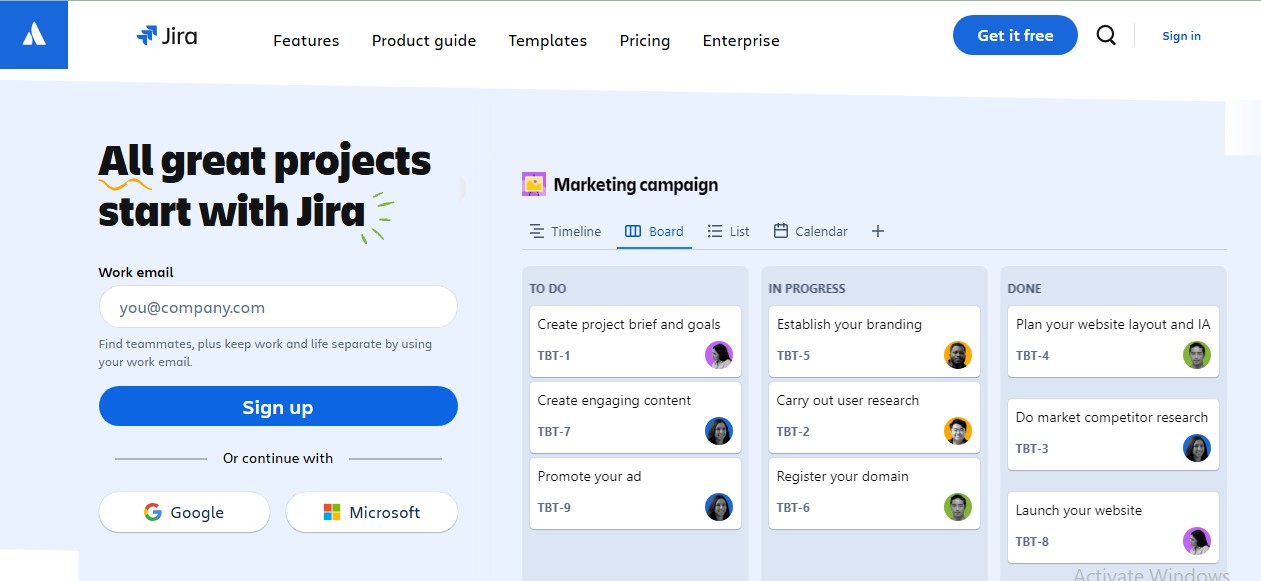
**Creating an Atlassian Account**

* Go to the signup page of Atlassian and enter the required user credentials asked on the site and then

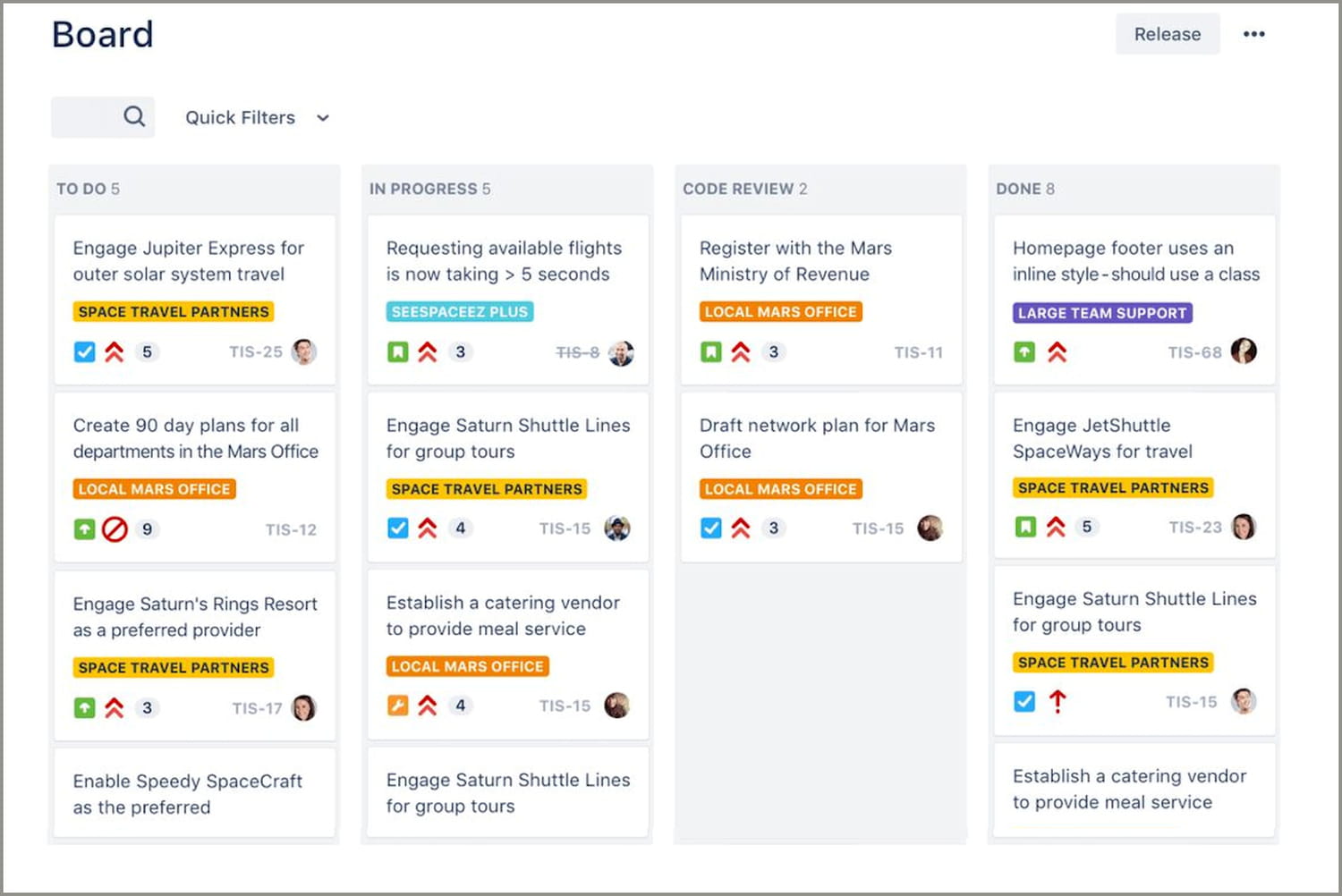
click on the Sign-Up button.

* To complete the setup and login, click the verification link in the email box.
* Set up your Atlassian account.
* Click on signup after filling details. The account has been created. The user is automatically redirected

to the home page.



**Interface of JIRA**



**1. Create**: You can either create a new epic in agile either use the issues you have created

JIRA board

**2. Plan:** Plan mode display all the user stories create for the project

**3. Dashboard**: Dashboard is the main display you see when you log in to Jira. You can create multiple

dashboards from different projects. ΟR multiple dashboard one massive overview of all the mark involved with.

**4. Project**: A project is a simple collection of issues. You would typically you a project to represent the

development work for a product, project or services in Jira software,

**5. Roadmap**: Road map in Jira software or service are team-level road map useful for planning large

pieces of work several months in advance at the Epic level within a single project. Dimple planning and

dependency management features help your teams visualize and manage work better together.

**6. Backlog**: The backlog view lists issues that your team plans to work on (and the backlog. Sprint lists)

as well as the issues currently your team's board on you use the backlog issue list to plan work in advance

so that team members can quickly jump tasks when they're ready.

**7. Active sprint**: The Active sprints of the issues that you’re a you’re on the most Scrum board

displays team is currently working on. You can create and update issues, and drag and drop issues to

transition them through a workflow.

**8. Reports**: Reports in Jira software are offers critical insights for scrum, Kanban, and any agile

methodology in between sprint report burn down chart, release burn down, velocity chart. Control charts

etc.

**9. Issues**: A Jira work in issue represents a single piece of a project, Time tracking allows! Teams to

record the amount of time they spend working on issues. Teams can create custom workflows to drive the

progression of issues.

**10) Releases**: A scrum or Kanban board. Releases represent paints in time for your project. They can

be used to schedule how Features are rolled out to your customers! a way to organize work that has been

completed for the project.

**11) Project Settings**: Once, You have created a project, you can Configure it to suit the needs of your

team or to adopt to a new piece of work. Your team Can use Jira projects to coordinate the development

product, track a project, manage a help of a desk, and more

**WEEK-03**

**1). Play and Act Agile Ceremonies:**

**INRODUCTION**

The Domino's app is a convenient mobile platform designed for customers to browse

menus, customize orders, and have their favorite Domino's food delivered or ready

for pick-up.

**Agile ceremonies:**

* Agile sprint ceremonies are essential components of a successful agile project.
* They are regular meetings that are held throughout the project's development cycle to ensure that goals
* are being met and that all team members are on the same page. The purpose of these meetings is to review progress, discuss issues, and make decisions.
* Sprint ceremonies also provide a platform for team members to collaborate, share ideas, and learn from

each other. They are a key part of the agile development process and involve all stakeholders in the project.

**Features:**

* Step1:Splash Screen
* Step2: Login/Signup Screen
* Step3: Home screen: Central hub for all features.
* Step4: Menu Screen: Displays all food options.
* Step5: Customization Screen
* Step6: Cart Screen
* Step7: Checkout Screen
* Step8:Real-time updates on the order status.
* Step9:Profile/Account Screen
* Step10:Settings Screen
* Step11:Payment mode
* Step12: payment success

**PROCESS:**

**Product Backlog:** Product Backlog is a prioritized list of all desired features, enhancements, bug fixes,

and other requirements for a product. It serves as the single source of truth for what needs to be built and

is managed by the Product Owner.

**Sprint Planning:** The Sprint Planning session is a meeting held at the beginning of each sprint to

review the goals of the project, assign tasks, and create a plan for achieving those goals.

**Sprint Backlog:** The set of Product Backlog items selected for a sprint, along with a plan for delivering

them. Created by the Development Team during Sprint Planning.

**Daily Stand-up (Daily Scrum):** The Daily Scrum is a meeting held every day of the sprint,

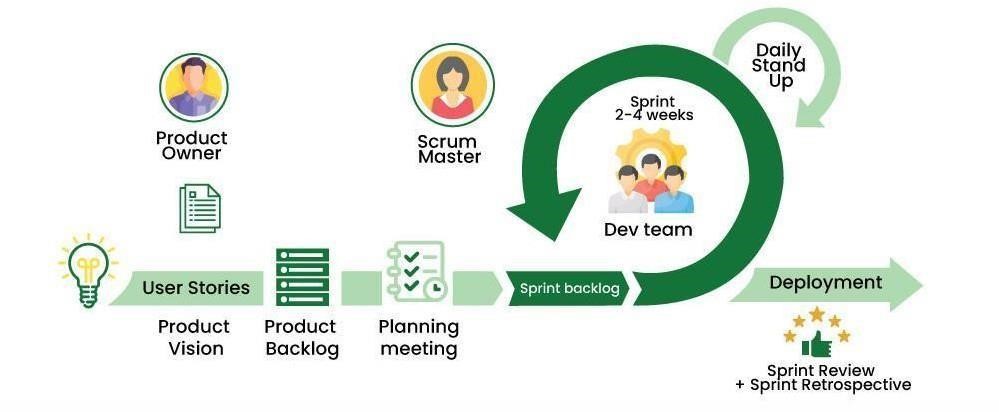
where team members discuss the progress they have made on their tasks and any issues they are facing.

**Sprint Review:** At the end of the sprint, the team holds a review meeting to showcase the work that

was completed during the sprint. The team demonstrates the work that was done and receives feedback

from stakeholders. This meeting is also an opportunity for the team to reflect on what went well and what

can be improved for the next sprint.



**Sprint Retrospective:** This is a meeting that is held at the end of each sprint and is used to reflect on what went well and what can be improved. The team discusses what worked well, what didn't work well, and what can be done differently in the next sprint. This meeting is led by the Scrum Master and is an opportunity for the team to improve their process and workflow.

**Sprint Refinement:** Held halfway through the sprint to review the tasks of the sprint and make any

necessary adjustments, with the goal of ensuring that the sprint is running smoothly and that all tasks are

completed on time.

**Repeat:** The process repeats with the start of a new sprint, beginning again with Product Backlog

Refinement and continuing through Sprint Planning, the Sprint, Sprint Review, and Sprint

Retrospective.

**Increment:** The sum of all Product Backlog items completed during a sprint, potentially shippable and

usable.

**2). Play Different Agile Roles Eg. Product Owner, Business Analyst**

The two key pillars for a successful agile project are the Product Owner (PO) and the Business Analyst.

**PRODUCT OWNER:**

A product owner is a role on a Scrum team that is responsible for the project's outcome. The product owner seeks to maximize a product's value by managing and making product best and effective.



**ROLES AND RESPONSIBILITY OF A PRODUCT OWNER:**

**1. Defining the vision:** The agile product owner is the person on the product development team, using

their high-level perspective to define goals and create a vision for development projects.

**2. Managing the product backlog:** One of the most important responsibilities for a scrum product

owner is managing the product backlog.

**3. Prioritizing needs**: Another key role of the product owner is to prioritize needs. In other words,

they must prioritize the triangle of scope, budget, and time, weighing priorities according to the needs and

objectives of stakeholders.

**4. Overseeing development stages:** With the vision, strategy, and product priorities set, the product

owner should spend a significant amount of time overseeing the actual development of the product. They

are a key player throughout each event, including planning, refinement, review, and sprint.

**SOFT SKILLS NEEDED TO A PRODUCT OWNER:**

• Communication

• Technical Skills.

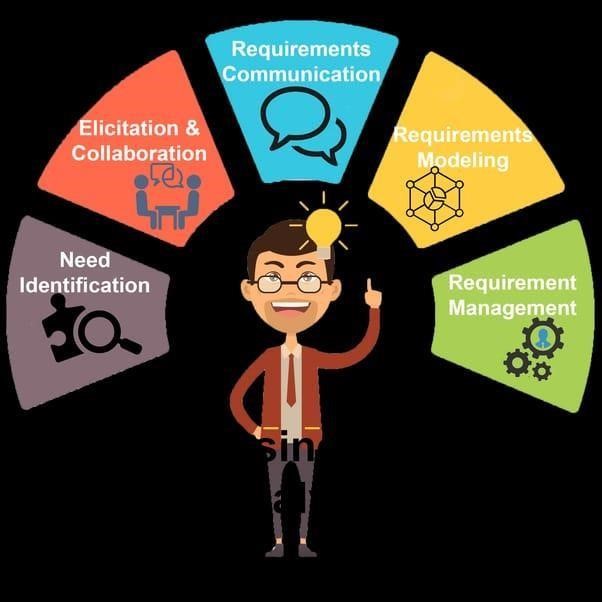
• Decision-Making.

• Project Management Skill.

• Collaboration

**BUSINESS ANALYST:**

Agile business analyst is about increasing the delivery of business value to the stakeholders of the project or product being developed.



**ROLE OF A BUSINESS ANALYST IN AGILE:**

• Maintain the focus on business value

• Identify missing requirements

• Coach the Product Owner (on work)

• Coach the development team (on business domain)

• Help define acceptance criteria before work starts

• Solving Business Problems

• Looking for Savings and Efficiencies

• Focusing on Business Development

• Performance Analysis

• Competitor Analysis

**SOFT SKILLS OF A BUSINESS ANALYST ARE:**

Analytical thinking and Problem Solving-solving, creative thinking, systems thinking, learning and

decision making

**1. Behavioral Characteristics** -ethics, personal organization and trustworthiness.

**2. Business Knowledge-**business principles & practice, industry knowledge, organization knowledge and solution knowledge.

**3. Communication Skills-**oral communication, written communication and teaching.

**4. Interaction Skills-**facilitation & negotiation, leadership & influencing and teamwork.

**5. Software Applications-**general purpose applications and specialized application.

**Week-04:**

**1).Case Study to understand the Importance of Risk Management and Mitigation of**

**Risk.**

**Telegram:**

Telegram Manager, commonly known as Telegram, is a could-based, cross-platform, encrypted instant messaging (IM)service. It was originally launched for ios on 14th August 2013 and Android in October 2013.it allow users to exchange messages, share media and hold private and video calls, as well as public live streams. It is available for Android, ios, window, macOS, Linux, and web browser. Telegram also offers End-to-encryption in voice and video calls, and in voice and in optional private chart, which Telegram calls secret chats.



**Problem faced by Telegram:**

1. **Poor data security:**

If telegram fails to implement robust-encryption and security measures, user data such as message, media file, and personal information could be vulnerable to unauthorized access, interception, or hacking attempts

1. **Out data System:**
2. **Performance issues:** out dated systems may struggle to run the latest versions of Telegram efficiently,

leading to slower performance, crashes, or compatibility issue.

1. **Limited supports:** Telegram may discontinue supports for out dated system, meaning user on those platforms well no longer receive assistance or updates vulnerable to unresolved issued issues or technical problems.

**3)Phishing and social engineering attacks :**

cycle criminals mays attempt to trick user into disclosing sensitive information on or installing through phishing attack conducted via telegram message or channels

**What are the disadvantages of Telegram?**

Not all of them will apply to everyone, but it's good to know the weaknesses in each app you rely on.

* Telegram Chats Aren't End-to-End Encrypted by Default.
* Telegram Collects Your Contact Data.
* Telegram Has Limited Support.
* Your Friends Might Not Use It.
* Ads and a Subscription Plan.
* Miscellaneous Telegram Concerns.

**Solution to this problem:-**

**1)enabling two-step verification:**

User can enable two-step verification in telegram app’s setting. This typically involves selecting a password and providing on optional recovery email address. once enabled, users will be prompted email

Their chosen password whenever they log in to their telegram account on a new device.

**2)use a strong password:**

Choose a strong unique password for tour telegram account, avoid using easily guessable password and consider using a password manager to generate and store complx password securely

**3)keep the app updated:**

Regularly update the telegram app to the latest version to ensure you have the latest security patches and features.

**4)secure you recovery email:**

If you’ve set up a recovery email for two step verification. Ensure it’s secure and accessible only by you.

Signal remains the top choice for privacy-conscious users. It uses end-to-end encryption by default, ensuring that only you and your intended recipient can access your messages. Signal collects virtually no user data and features an open-source codebase, which allows independent security experts to verify its safety.

**02). How to use tool to manage and mitigate risks [e.g.: logic gate, board]**

**Logic gate:**

It is a software where you keep all the information is in one system (logic gate).it includes

policies, goals& priorities.

Or

Logic gate’s controls management solutions gives your risk and control owners a break from

the disorganized mess of spreadsheet email and documents

**Working:**

1. It will work from single central hub.

2. It will according to the industry standard control frameworks (ISCF).

**For example:**

Management wants task ‘E” to be done on priority but the employee gets the email

after 2 days about priority.

**Advantage:**

By using logic gates everyone gets information at real time.

**1. Emails:**

Logic gates can be used for emails as well.

2. If we find any risk then we can plan &everyone can work on real time.

It’s a rapid session where team will monitor, control assessments, gather evidence

and much more, right in the platform. They will continuously collaborate and share

information such as tracking, finding or triggering automatic emails when an automatic

assessment is kicked off.

**Week-5**

**1). Conduct warm up activities to ignite Design Thinking**

1. **Puddle jumper:**

**m**

### ****Equipment Needed for Puddle Jumper Minute to Win It Game****

* 6 Full Cups of Water per person or team
* Ping Pong Balls per person or team

## ****How to Play the Puddle Jumper Minute to Win It Game****

1. Prior to game start, place the full cups of water in pairs at a slight distance apart.
2. When the clock begins, the player places a ping pong ball into the first Solo cup and attempts to blow it

On to the Solo cup across from it.

1. To complete the game, blow a ping pong ball onto each of the full Solo cups within the 60 second time

limit.

4.Each player is faced with two rows of cups and as many columns as desired. The cups are filled with water, and a ping pong is balanced on the row nearest to the players. The players must then blow on the ping pong, make it leap from one cup to another. If the ball falls on the ground, the player has to try again.

### ****The Rules of Puddle Jumper Minute to Win It Game****

1. Only one ball may be placed at a time.
2. The player may not touch the ping pong balls

### ****Target cups****



**Requirements**

1)paper cup

2)plastic rubber band

3) floated paper

**Procedure:**

In this game 2 member can participate in from the team, in front of the participants we keep 3 to 4 cup of pyramid.

we tell the player to fall the paper cup pyramid using rubber and folded paper and we start the timer to check how much player has taken to fall the paper cup pyramid

**Rules:**

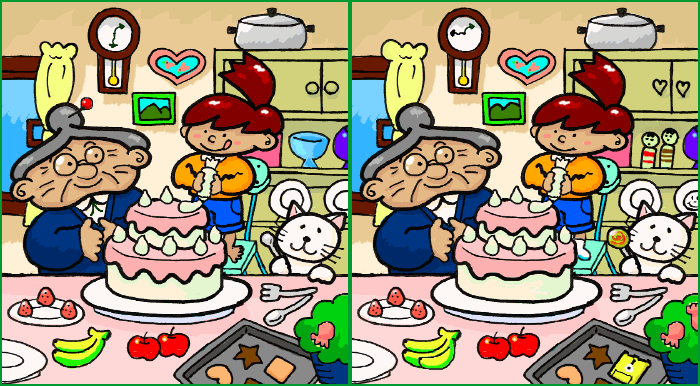
* Don’t cross the line
* All the cup are counted
* Another player can build the pyramid when its fall down
* Winner is announced when the team is taken the less time to clear the 3 set of pyramid that team is the winner**.**

**2). Organize and Conduct Design thinking exercises and games:**

**Design Thinking:**

Design thinking is a non-linear, interactive process that teams use to understand users, challenge assumptions, redefine problem and create innovation solutions to proto type and text.

**A. Finding the difference**



**Uses:**

* Improve concentration
* Helps the person to think

**Requirement:**

* Difference image
* Timer
* Whistle

**Setup Instructions**

1. Divide the group into two teams. Five or six players per team is a good size. If you have a large group, you can run several games at the same time.
2. Choose one team to line up across the front of the room. This is the “lined-up team.”
3. The other team will be the “observing team.” They will stand where they can clearly see the lined-up team.

**Procedure:**

We can separate the team and give to whistle to team and start to display the difference image on start to display the difference image on the projector. Now start to count 10sec and tell to which team knows the difference blow the whistle team and come to front and show the difference. When we give another opportunity to other team to find other different remains first team finds the difference

**Rules:**

* Don’t blow the whistle before 10sec.
* One person blow whistle in a team.
* One person come and show the difference.
* 1min is given to shows the difference.
* Which team find the most difference that team win this game.

**Benefits of Solving Spot The Difference Exercises**

* These enhance attention to detail.
* Upscale observational skills.
* Improve visual perception.
* Promote brain activity.
* Boost mental sharpness.
* Reduce cognitive fatigue.

**B) Act and show**



**Uses:**

* Improve concentration
* Help the person to think

**Requirements:**

* Timer
* Words

**Procedure:**

* We can separate the team member and start writing the batch name in small paper and fold the miss

paper and call the team name.

* give the one name to the member in the team and start the time and tell to do the acting we give the first

opportunity to guess the word.

* if the team is fail to answer or guess the next opportunity is give to the next team to guess the word.

**Rules:**

* Don’t use your lip to act
* Don’t make the action like, writing letters.
* After 1min if that is guesses the correct answer it is not counted.

**Week-06**

**Organize role play for requirement activities**

**Participants**

Prakruthi

Ragini

**Role play script**:

Prakruthi: Hey! Ragini, have you applied for your ration card yet?

Ragini: No, I've been meaning to, but I haven't had the time. How do I do it?

Prakruthi: You can apply through Karnataka One Services. It's really easy!

Ragini: Okay, cool! What do I need to do?

Prakruthi: First, you need to download the Karnataka One Services app or visit their website.

Ragini: Done! What's next?

Prakruthi: Next, you need to click on "Apply for Services" and select "Ration Card".

Ragini: Okay, got it! What documents do I need to upload?

Prakruthi: You'll need to upload your Aadhaar card, income certificate, and residence proof.

Ragini: Alright! And how do I pay the application fee?

Prakruthi: You can pay online using your credit/debit card or net banking.

Ragini: Cool! Thanks for guiding me through this,Prakruthi!

Prakruthi: No problem, happy to help!

**App requirements**

* Easy Registration:
* Service Catalog:
* Personalized Dashboard:
* Simple Navigation:
* Accessibility Features:
* Notification System:
* Feedback Mechanism:
* Security and Data Privacy:

**2.Identify a problem and prepare requirement document or epic and user stories**

**Tasks**:

* Create user registration form
* Validate user credentials
* Send verification email
* Activate user account

**User story 2: Service Catalog Tasks:**

* Create service catalog page
* List available government services
* Provide service descriptions and eligibility criteria

**User story 3: Service Application Tasks:**

* Create service application form
* Validate user input
* Attach required documents
* Submit application

**User story 4: Application Tracking**

**Tasks**:

* Create application tracking page
* Display application status
* Provide updates on application processing

**User story 5. Payment Gateway**  **Tasks**:

* Integrate payment gateway
* Validate payment information
* Process payment
* Provide payment receipt

**User Story 6: Search and Filter Services Tasks:**

* Search Bar Design
* Filter Option Implementation
* Search Algorithm Development
* Search Functionality Testing

**User Story 7: Service Application Status Updates Tasks:**

* Status Update Design
* Notification System Implementation
* Application Tracking Integration
* Notification System Testing

**User Story 8: Online Payment History Tasks:**

* Payment History Design
* Payment Data Development
* Payment History Implementation
* Payment History Testing

**User Story 9: Service Feedback Mechanism Tasks:**

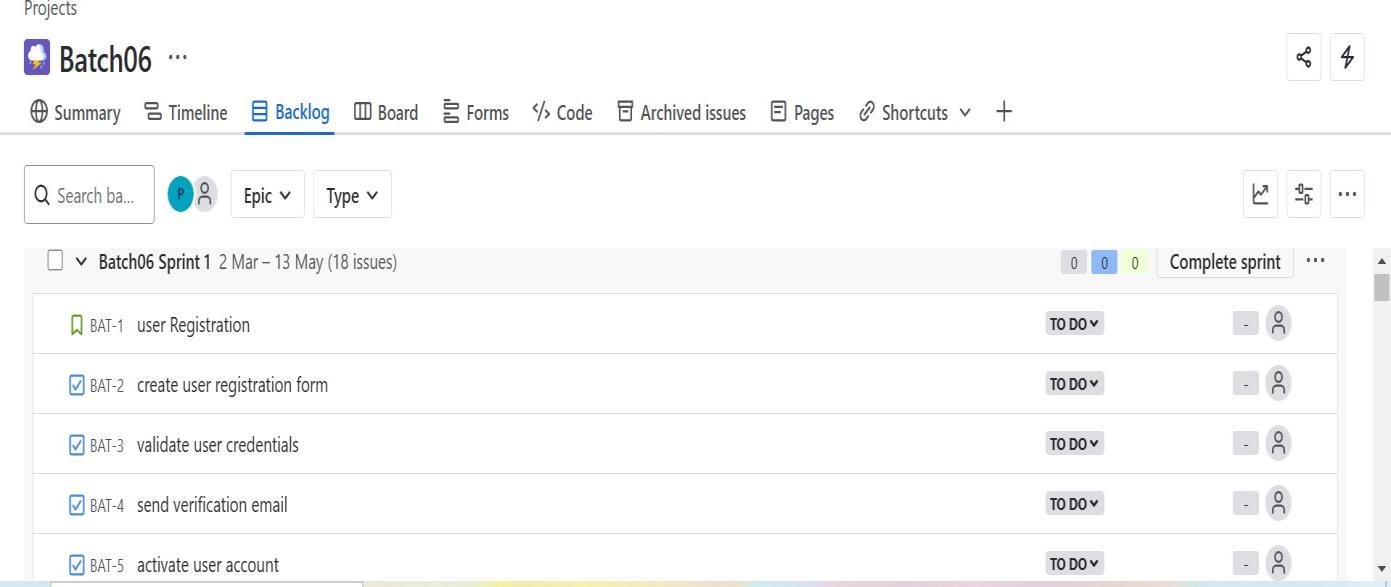
* Feedback Form Design
* Feedback Submission Implementation
* Feedback Analysis Development
* Feedback System Testing

**User Story 10: Multi-Language Support Tasks:**

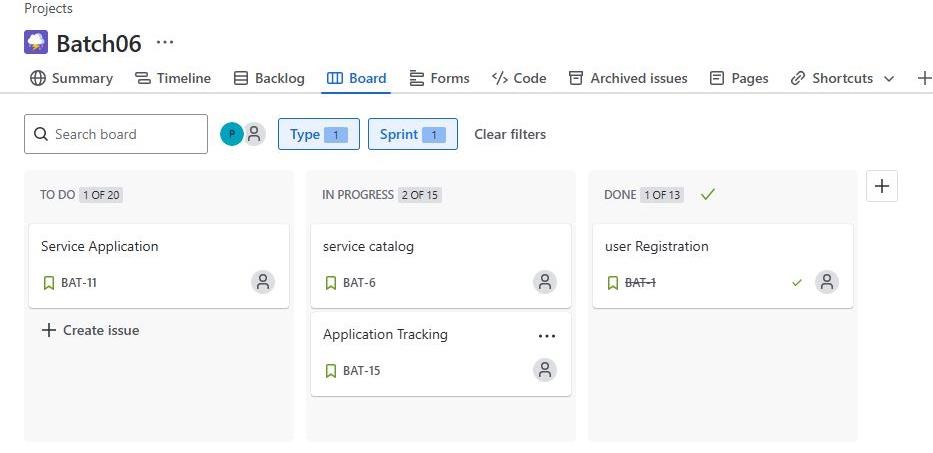
1. Language Selection Design
2. Language Translation Implementation
3. Multilingual Support Development
4. Language Support Testing

**3.Configure JIRA for the managing the project to solve the identified problem**

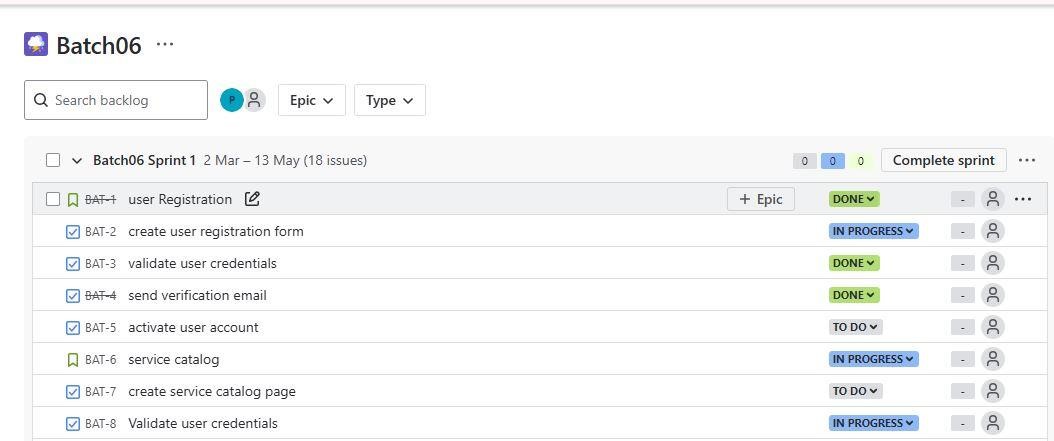
**BACKLOG**



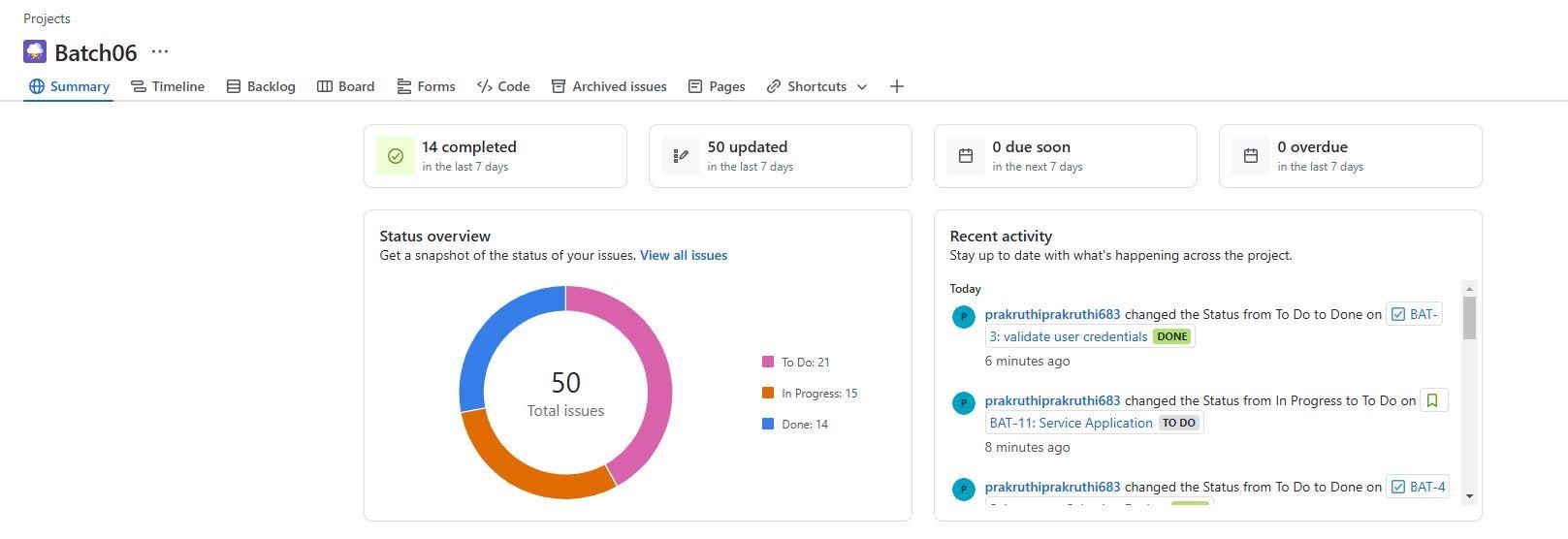
**BOARD**



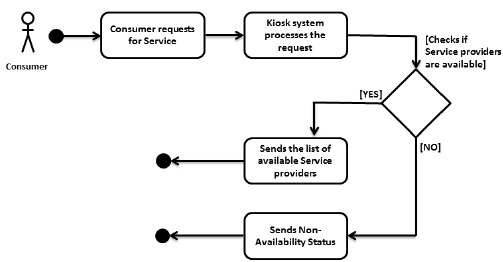
**BACKLOG**



**SUMMARY**



4.Draw UML diagram for given user case



**Week 07**

* 1. **Create detailed user stories for the identified problem**

**Epic : Karnataka one services**

**User Story 1: Register for Karnataka One Services**

As a citizen of Karnataka, I want to register for Karnataka One Services, so that I can access various government services online.

**Acceptance Criteria**

1. The system should allow me to register using my Aadhaar number, email, or mobile number.

2. The system should verify my identity using my Aadhaar number or other government-issued ID.

3. The system should create a unique username and password for me.

4. The system should send a confirmation email or SMS to me after registration.

**User Story 2: Apply for Services**

As a registered user of Karnataka One Services, I want to apply for various government services, such as ration card, driving license, or passport, so that I can access these services online.

**Acceptance Criteria**

1. The system should allow me to select the service I want to apply for.

2. The system should display the required documents and eligibility criteria for the selected service.

3. The system should allow me to upload the required documents and fill in the application form.

4. The system should verify the uploaded documents and application form.

5. The system should display the application status and allow me to track the progress.

**User Story 3: Track Application Status**

As a registered user of Karnataka One Services, I want to track the status of my application, so that I can stay informed about the progress.

**Acceptance Criteria**

1. The system should allow me to view the status of my application.

2. The system should display the current stage of the application process.

3. The system should send notifications to me via email or SMS when the application status changes.

**User Story 4: Make Payments**

As a registered user of Karnataka One Services, I want to make payments for various government services, such as application fees or utility bills, so that I can access these services online.

**Acceptance Criteria**

1. The system should allow me to select the payment method (e.g., credit/debit card, net banking).

2. The system should display the payment amount and payment instructions.

3. The system should allow me to make the payment securely.

4. The system should display the payment receipt and update the application status.

**User Story 5: View Dashboard**

As a registered user of Karnataka One Services, I want to view a dashboard that displays my application status, payment history, and other relevant information, so that I can access my account information easily.

**Acceptance Criteria**

1. The system should display a personalized dashboard for me.

2. The system should display my application status, payment history, and other relevant information.

3. The system should allow me to customize the dashboard to display the information I need.

**2.Organize and play planning poker to decide on user points:**

Planning Poker is a popular agile estimation technique used by teams to estimate the relative complexity of tasks, user stories, or features. It's a fun and collaborative way to estimate effort, time, or complexity.

**Planning Poker on User Story:**

Planning Poker on User Story is a collaborative estimation technique used in Agile software development to estimate the effort required to complete a specific User Story. Team members anonymously select cards with numerical values to represent their estimates, and then discuss and reconcile any discrepancies to arrive at a consensus estimate for the User Story.

Estimation User pointer

|  |  |
| --- | --- |
| Registration for Karnataka-1 service | 5 |
| Apply for services | 10 |
| Track application status | 20 |
| Make payment | 15 |
| View dashboard | 10 |

**Reasons for the user pointer and estimation**

**User Story Pointers:**

1. Integration Complexity: Integrating multiple government services and departments may increase pointer values.

2. Data Security: Ensuring the security and privacy of citizen data may require higher pointer values.

3. Scalability: Designing the platform to handle a large volume of users and transactions may increase pointer values.

4. User Experience: Creating an intuitive and user-friendly interface for citizens may require higher pointer values.

5. Compliance: Ensuring compliance with government regulations and standards may increase pointer values.

**Reasons for Estimation:**

1. Service Coverage: Estimating the effort required to integrate multiple services and departments.

2. Technical Infrastructure: Estimating the resources needed to develop and maintain the technical infrastructure.

3. Citizen Engagement: Estimating the effort required to promote the platform and engage citizens.

4. Data Migration: Estimating the effort required to migrate existing data to the new platform.

5. Training and Support: Estimating the resources needed to train government officials and provide citizen support.