Building a game wandering in the woods

Software Engineering (Sections SU22-CPSC-60500-001 & 002)
Instructor's name: Fadi Wedyan

Team members:

Raj Kumar Reddy Yerra
Sampath Bandla
Sairam Cheruku
Sai Teja Reddy Chittepu
Hemanth Ciripuram

Table of Contents

Abstract 3

1. **Introduction:** 3
2. **Roadmap:** 3
3. **Vision:** 4
4. **Scope:** 4
5. **Agile Software Development** 4
   - **5.1 Principles of Agile development** 5
   - **5.2 Agile development in our application** 6
6. **Project Requirements** 7
   - **6.1 Scenarios** 7
   - **6.2 Personas** 8
7. **Features of cloud application:** 8
8. **Use Stories for cloud application:** 9

9. Mobile Development Stack 11

10. Infrastructure for cloud Application 12

11. Data Model for Cloud Application 12

12. Application Architecture 13

13. Security and Privacy Considerations 16

14. Testing strategy 16

15. Branching and deployment strategy 18

Abstract

The project is about building a Game title "Wandering in the woods". It's a mysterious game where people will wander and an AI agent at the opposite will play with the players to produce some interesting unusual results. At every move a splash will be replaced in the grid box.

1. Introduction:

This game is built for different grades students with different levels of understanding and playing. For k-2 grades students it's made easier just displaying grids and two players will start diagonally moving from corners and both players will collide to end the game with a happy message.

For grades 3-4 students there can be 3 to 4 players and it's a next level instruction stage where they can place them in the grids and statistics will be showed in the right corner of all players.

For grades 6-8 students level some more advance level is modified where students will be able to modify grids.

2. Roadmap:

Following is the road map for our application:

Identification

- Technology choice (software, hardware)
- Identification of infrastructure components.
- Evoke implementation strategy.

Infrastructure setup

- Identify service catalog offerings
- Infrastructure setup and assessment

Architecture and development model

- Identify development model with various components from operational model
- Infrastructure backup and recovery process
- Installing binaries and getting them ready

Environment build

- Performing pre installation checks
- Installing and configuring cloud components
- Provision and deactivation of use cases

Advance features and deployment

- Usage monitoring
- Performance monitoring
- Deployment of live project

3. Vision:

Vision of this project is to give students and understanding that how to build games with interactive interfaces and hands on experience of Graphical screens. Like in this game we build Welcome screen, game on screen, game over screen.

4. Scope:

This game is especially for k-grades, 3-4 grades and 6-8 grades students. To give them a happy playing and mysterious game which make them curious about the next level.

5. Agile Software Development

An agile software development is a form of development process in which development process is break down into chunks and each individual functionality is accomplished separately. Agile development methodology was developed in contrast with 'Waterfall

model'. As waterfall approach takes long time to deliver long products. And if requirements are to be added it will result in wrong product. But agile method is a responsive behavior in which customer and developer is being connected and a quick response is shown from development team regarding changes or adding new functionalities in the application. Agile method maximizes the quality of products and minimizes the time of development.

Agile development methodology is based on team work. All the tasks are divided into respective teams of filed and accomplished in an order. At the end a quality product is generated.

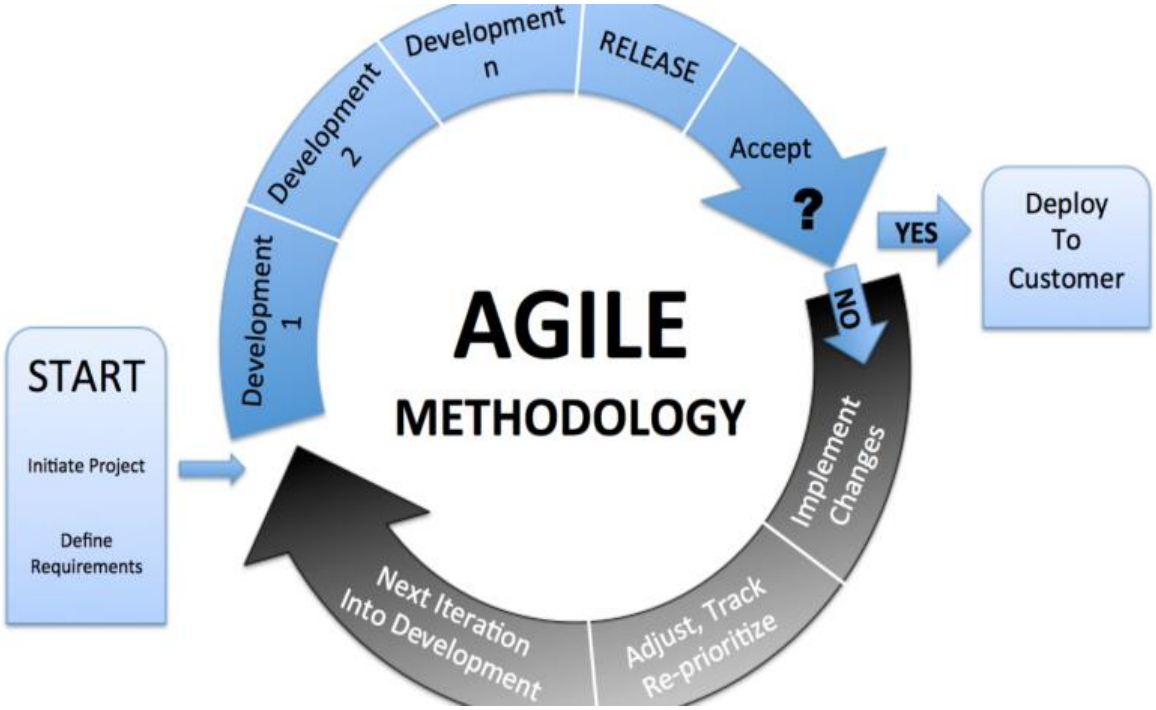

Figure 1: Agile Process

5.1 Principles of Agile development

- Customer satisfaction by delivering product quickly.
- Changing the requirements at any stage of development process or event after the development.

- Deliver valuable product
- Application must be in working state as it depicts the progress.
- Maintain a sustainable working pace during development process.
- Monitor the team work for better and valuable progress.

5.2 Agile development in our application

The following entire agile methodology process (step by step) will take place while developing mobile phone application:

Automation of end-to-end process

- Plan user or project admin and Task hierarchy
  - Code the tracker integration
  - Track bugs, artifacts and requirements
  - Build and test integrated components
  - Run the developed component

Team Collaboration

- Rea time reports and status
- Discussion forums
- Projects based Wiki
- Indexed objects and file types

Social Architecture

- Community and projects
- Workspace

User Tool Integration

- Business users
- Developers
- User point Tools

- Life cycle tools
- Code analysis
- Unit testing
- Planning
- Report management

6. Project Requirements

Some of the requirements for cloud system projects are below:

- **Providing Multi-tenancy :**
  Multi-tenancy eliminates the problems which were created by the traditional software's in models and licensing.

- **Regular delivery**
- **Integration on demand**
- **World class data center and security**
- **Sustainable infrastructure**

6.1 Scenarios

The scenarios of the use of the game application include;

k-2 students

- In this scenario the students will be given a simple interface of game with fix grid width. Both players will start playing by moving diagonally and their score will be counted on each move. When they bump game will end.

3-5 students

In this scenario an advance level of game will be displayed where grid width and shape will be changed and students can play game multiple times.

6-8 students

This scenario includes most advance level of game where students can select any number of grid size.

6.2 Personas

The personas that will use the Game application include;

- The IT decision maker(includes cost savings and operational efficiency decisions)
- The developer(develop, test and deliver quality product)
- The big data operator(data management solutions)

7. Features of Game application:

1. **Resource pooling:**
2. **On demand self-service:**
3. **Easy maintenance:**
4. **Security:**
   It provides high security regarding loss of data or stealing by hackers.
5. **Availability:**

8. Use Stories for cloud application:

User Story 1:

As a **New User**, I want to access play application.
Acceptance Criteria • New User have to download the application in order to play.

9. Mobile Development Stack:

The term "mobile development stack" refers to the set of programming languages, tools, frameworks and other technologies needed to create a working mobile app. This game will be developed using ‘Arcade’ library of python which is used for creating Graphical interfaces of games.

The mobile stack can be divided into following components:

a) **User:** Persons who will use this mobile application.
b) **Frontend:** To create the interface through which end-users can interact, frontend technologies are used. Frontend will be developed using Arcade (python library)

Following mobile development stack is used for our project:

User	Frontend
Persons	Arcade

10- Infrastructure for Game Application:

Hardware:

This application will be installed on mobile phones and can be accessed through web online system.

Operating System:

Its android version can only be installed on android devices (mobile phones), and iOS version can only be installed on apple devices (iPhones).

Development:

It is developed using Arcade as front-end.

11- Security and Privacy Considerations

Software security is a major problem that must be tackled in order to create reliable software systems. In order to secure user's data and information, a strong data encryption will be utilized to keep it safe in database. In the same way, on cloud data will also be in encrypted form.

12- Testing Strategy

Two types of testing will be performed on this application.

1- Functional Testing
2- Non-functional Testing

1- Functional Testing:

In functional testing, we will test the app with following testing methods.

- Unit Testing

- Integration Testing
- System Testing
- Acceptance Testing

Unit Testing:

Unit testing will be performed in development process to test each component of application. Each feature of application will be tested like buttons are working correctly, input fields are taking input and where the input data is going, components on screen are placed well etc.

Integration Testing:

After completing unit testing on each component, units are then integrated with each other to create a module. In integration testing, each module is tested either units are integrated correctly or not. For example, input fields and buttons are used in a form to create login or registration form, data sent from application to cloud is being sent correctly and saved.

System Testing:

It is a black box testing, in which the system as a whole will be tested. A separate testing team will test the system. User registration, taking phone backup on cloud, restoring old phone data from cloud will be tested.

Acceptance Testing:

In acceptance testing, the final system will be tested either it is ready for use or not. It will be ensured that it works on each phone correctly or not. It will be tested be some users to get feedback either it is ready or not.

2- Non-functional testing:

In non-functional testing, we will test the app with following testing methods.

- Performance Testing
- Security Testing
- Usability Testing
- Compatibility Testing

Performance Testing:

The performance and responsiveness of the app will be tested in performance testing.

Load Testing: In this process, multiple simulations will be performed at a time to check the performance of the app. Multiple restoring and backup processes will be done at a time to check if app responses fast or not.

Stress Testing: In stress testing, the load will be increased to check how much users can this app response fast at the same time.

Endurance Testing: The behavior of the application is tested under overloaded simulations.

Spike Testing: is a form of load test intended to see how your software will handle much larger bursts of concurrent user or system activity for varied periods of time. This should give you a better idea of what will happen if the load is suddenly and significantly raised.

Security Testing:

With the rise of cloud-based testing platforms and cyber threats, the security of data used and stored in software is becoming increasingly important. The encrypted data will be saved on cloud to secure the data.

Usability Testing:

The purpose is to verify whether an application's apparent design and visuals suit the planned workflow for various tasks, such as logging into it. Usability testing is a wonderful approach for teams to see if certain functions, or the entire system, are easy to use.

Compatibility Testing:

In compatible testing, the application will be check its compatibility to run on different mobile phone.

13- Branching and deployment strategy

Branching Strategy:

We will use branching strategy for this application to maintain and keep our application up to date in future. After testing of application, it will be our first version of application. In future if any changes are needed, new branch will be created to keep the older version safe. In new branches the updated versions of application will be developed. This is a good way of maintaining the versions of application and updating versions time to time.

Deployment Strategy:

"A/B Testing" will be used for deployment strategy. In this strategy, we will test a new version on a limited servers and will redirect some traffic to it. It means we will be doing some tests on a variety of ideas to discover which one works best.

14- Feedback on project Experience

The project was about to write a software requirement specifications (SRS) for a cloud application that can provide cloud storage to nonprofit organization to store their data and increase work experience virtually by minimizing employer and work load. We have learned how we can understand all about the project before going to its implementation to avoid any failure of application. The tricky part was application architecture in which we have to build Data Flow Diagram, Use Cases etc.