Wednesday, 29 November 2017

ICTPRG529

Software Management Tools

- Software management tools
  - Project Management
    - Jira ~ https://www.atlassian.com/software/jira
    - A clean and prevent collaboration tool allowing multiple teams ranging in disciplines to communicate and submit tickets
  - Source Control ~ https://git-scm.com & https://github.com
    - Git
    - A command line / terminal based source control that is commonly use among the it industry. Highly efficient and allows for quick reverts, updates…
    - Git can be stored locally, on the cloud or on a local / company server. Or all 3 to avoid delays / issues during development.
  - Collaboration Management
    - Asana ~ https://asana.com/?utm_source=capterra&utm_campaign=project_management&utm_medium=ga
      - Another highly popular management tool, can be used via browser or downloaded allowing for a quick and secure link for either in house or remote submissions
    - Trello ~ https://trello.com
      - A relative competer in management.

Methodology

Agile ~ Works well with my software management tools, also is the most modern and allows for a modular like development process that involves the whole team/'s <could expand what agile is >

Project Plan

The project is to build a affective, functional and stable app for mobile devices that will enable viewing, playing, receded of scores.

We will need a multiple teams working together across multiple areas i.e. Artist / Graphical, Engineers, Business Managers for financial and marketing to ensure that the app is know about

This project would involve a number of smaller steps / checkpoints. This is a modular approach that would enable all teams to come together and decide what needs work, improvement scrapping etc.

This is idea is built into the agile development, and heavily aided by the git source control software and Jira.

<u>Work Flow</u>

1. Research of application idea / Scope
2. Preliminary design and prototyping of app and various ideas
3. Graphical designs along with engineer designs for software
4. Review and sign off of design, scope and software deigns
5. Begin initial setup and structor of application
6. Reviewing and testing app using TDD
7. Will be broken down into 5 main steps
    1. Frame works build
        1. Either build or open source
        2. Tested
    2. App features build and implemented
        1. Num of App features
        2. Tested
    3. Graphics added
        1. By both Artis and engineers
        2. Tested
    4. Full app stress test

    1. Review and changed/ updated as needed

5. Released and updated
    1. Both from reviews and general improvements both technical and business related i.e. added feature to ensure a user base is maintained

The selected software above would enable both the technical teams and non-technical teams to communication via both individually or I groups. The stated software would also enable maximum productivity as it is coupled with an affective and proven methodology.

Detailed Review

During the project there were some considerable issues that were both rectified and or changed. Most of the issues were minor i.e. in-code-changes during individual team meetings. Most resulted in a small alteration from the initial pseudo code architecture.

Some issues cause multiple architectural changes mainly again in how the code / structor deviated from the initial design. However most changes were caught early and all if not most of the projects initial features have been implemented with adequate testing for stability.

During the project the project tools did aid in the overall work flow and to avoid issues surrounding the development work flow.

Pros:

A. Well designed software interfaces

B. Easy to add and modify tickets

C. High cohesion among submitting tickets -> solutions

D. Hight cohesion between dev team, management , business, graphics admins

E. In real time updates

F. Worked well when in the company eco-system

G. Allowed for an higher degree of details reporting and analyst

Cons:

A. Cost sizeable funds for the software alone,

    A. i.e. had to purchase business related packages for the software, usually coming in size ranges

B. Some required local eco-systems to be built and maintained

C. Required added stability to company hardware infrastructure as considerable more traffic and real time communication was needed by the tools.

D. Learning curve along with training courses cause delays and errors

E. Some technical bugs were discovered and caused delays, mainly with source control and merges among code bases

Agile ~ https://www.cprime.com/resources/what-is-agile-what-is-scrum/

~ Agile software development refers to a group of software development methodologies based on iterative development, where requirements and solutions evolve through collaboration between self-organizing cross-functional teams