# **Project Overview**

Presented by: Christopher Campbell(S1424416), James Hall(S1428914), Heather reid(S1430421), jack mcivor(S1427722), patrick lafferty(S1428450)

Group 13



Project Initiation

For integrated project 3

Our group consists of Christopher Campbell, James Hall, Heather Reid, Jack McIvor and Patrick Lafferty, and our group number is thirteen. As a group, we have decided to undertake the Document Management System. We all agreed that this project was best suited to the different skills that the group possess. The Document Management System which we plan to create will allow Ideagen Customers to log in and create documents which they can then share with other users of their choice.

We have discussed different ways in which we can tackle how to create the system, but decided that we will be using PHP and MySQL.

# **Roles**

## **Project Roles**

### **Scrum Master**

The Scrum Master is the coordinator for our agile development team. Scrum is a methodology that allows a team to self-organize and make changes quickly, in harmony with agile principles. The Scrum Master manages the process for how information is exchanged.

### **Project Manager**

The Project Manager is responsible for coordinating the internal resources and third parties for the flawless execution of projects. They create and maintain project documentation throughout, including a detailed project initiation to track progress and perform risk management to minimize project risks. For a project manager to be successful, they require to have solid communication skills, written and verbal, with clients and the team.

### **Product Owner**

The Scrum product owner is typically a project's key stakeholder. Part of the product owner responsibilities is to have a vision of what he or she wishes to build, and convey that vision to the scrum team. This is key to successfully starting any agile software development project.

### **Stakeholders**

A stakeholder is any person, organisation, social group, or society at large that has a stake in the business. Thus, stakeholders can be internal or external to the business. A stake is a vital interest in the business or its activities.

### **Development Team**

A self-organizing, cross-functional team of people who collectively are responsible for all the work necessary to produce working, validated assets is essential to creating the system.

## **System Roles**

### **Administrator**

The administrator is responsible for maintaining and effectively running the system. They can do this by adding/deleting/editing user account information and resetting passwords, etc. They will also be responsible for documenting the configuration of the system. As well as handling the front-end documentation, the administrator will also be responsible for maintaining the back-end database.

### **Document Creator**

A Document Creator essentially creates a document in which they can control via the system. With the document, they can; title it, change document status, add/edit revision number, add and remove distributees, add/remove attachments, etc.

### **Distributee**

A Distributee will view a document through the system, at the permission of the Document Creator.

# **Methodology**

As a group we have decided to follow an agile software development life cycle which is an iterative process. The life cycle we have choosen is Scrum. One of the main reasons we choose Scrum is that the project proposals are subject to change after the development begins. i.e. the stakeholder sees an initial plan for the development and realises that the project team have misinterpreted the requirements specification.

## **What is Scrum?**

In general Scrum is a series of steps that are repeated until a list of tasks are satisfied. Below is a short description of each stage:

1. The first stage of the Scrum process is when the Project Owner creates a wish list by prioritising the most important tasks.
2. The second stage is when the actual development takes place:
   1. Each wish is taking in turn and a Sprint backlog is created first
   2. Then the project team work on the task for 2-4 weeks whilst having daily meetings.
   3. In addition to the 2-4 week sprint there is also a daily Scrum sprint which takes place over 24 hours.
   4. By the end of the 2-4 week sprint a small part of the product should be shippable
   5. The work is then review and analysed, and the process is repeated until the project owners wish list is satisfied.

When the wish list is complete we hope to have a happy project owner and a viable product.

**Tools & Technologies**

To develop the document management system, we will create a web based solution using HTML, CSS, JavaScript JQuery and XAMPP to locally host our database. The reason for using such an approach is because all the group have previous experience and are familiar or comfortable with at least one aspect and would be happy to use it. To aid the development we will also be using a framework called Codeigniter.

Codeigniter will allow us to increase productivity by decreasing the amount of code we have to write as it comes with database access, caching, validation & authentication all built in, whilst having a small footprint. It also comes with much more features and plugins which we will probably discover and find useful throughout the project. There are many videos, documentation and a community forum to help fix any errors or problems which may arise during the project.

Version control will be a very important aspect during this group coursework as there are 5 members in our group. To tackle the issue of controlling versions we will use a tool called git. This will allow each member to track changes to the files and coordinate the work among each member and by the end we should have a full timeline of the development process which will come in useful when it comes to documenting the project and features implemented in the system.

# **Testing Strategy**

To ensure the effectiveness of the system, it needs to be tested thoroughly throughout the implementation period. This can catch the mistakes and errors, the faults and inefficiencies within the system to eventually perfect the project.

From using Scrum, the team will be working in Sprints. At the end of each Sprint will be an increment to the product that can be worked upon. However, at the end of each Sprint, this increment will be tested to see if it’s a reliable product to improve on, without any errors ruining the product from mistakes made in a previous increment.

To ensure software is as defect-free as possible, each single component of the system is tested. As we are using Scrum, a very agile approach to development, this method will focus on constant testing and revision of the system. Therefore, unit testing will fit in the development approach. The incremental approach will allow the testing team to only test the new additions on each increment without having to test the whole system at the very end of the development stage, only to uncover errors made very early in the implementation.

This encourages developers to modify the source code without immediate concerns about how such changes might affect the functioning of other units within the system or the whole program itself.

As this is not a very large system, unit testing is possible. Despite the process being tedious and time-consuming, unit testing covers every possible process within the system. As with all testing, these tests should be documented, making the process even more time-consuming.

Through testing, we will find a Minimum Viable Product(MVP) that can be delivered to the client despite not reaching potential perfection. It is important to have a MVP as soon as possible to ensure that the rest of the development time can be spent on improving upon the product. Constant testing is essential to getting the MVP early in the development stage.

# **RISKS**

There can be numerous reasons a risk can occur throughout the development and implementation of a project. Each risk will vary in severity and the probability that they may occur.

Below are risks that may occur for the development of the document management system. These risks can vary from within the project team, to the technologies that can be used.

The Risk Mitigation is used to help the development team understand and deal with potential risks, this can mean completely avoiding the negative outcome or simply reducing the effect it has on development.

|  |  |
| --- | --- |
| Threat | Risk Mitigation |
| Requirements Analysis - The requirements analysis created by the development team could be poorly written, vague or simply wrong. | Regular meetings with the stakeholders of the system to clarify any issues with the requirements. |
| Team Members Absent – There could be numerous reasons a team member could be absent throughout the development process. This may also range from the team member being permanently or temporarily out of the team. | If a member is absent or missing throughout development, there will have to be a fall-back in place. The leader should be willing to assign others to fill their workload. |
| Unexpected Risk - If a risk occurs that isn’t planned out, it can cause serious complications for the development team, ranging in severity. | Create a more detailed and thorough risk analysis that can cover most risks, that would have an effect  on the outcome of the system. |
| Unrealistic Schedule – The schedule provided may not be feasible for the whole of the project, due to a variety of issues that may arise from poor scheduling. | Using a suitable methodology with realistic milestones can help the development team produce a schedule that may result in the better management of the development teams time. |
| Poor Testing – If the testing for each increment is poor, this can result in errors/bugs slipping past the development team into the deliverables that are shown to the customer. | Making sure the testers know what they are doing/looking for and that they take the task seriously. Hiring the right testers or training them properly is key  for the success of testing. |
| Size of Software – The size of the project and the work required may be severely underestimated by the development team. This may result in deadlines being missed or the quality of the deliverables suffering. | By making sure the development team knows their tasks, follows the schedule and knows their limits to what can be done in a specific timescale. |

# **Milestones**

Additional features may be added to each milestone.

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone ID | Deliverables | Date Start | Date Completion |
| 1 | Create User, Edit User/Update User | 13/02/2017 | 24/02/2017 |
| 2 | Create Document, Activate Document, Revise Document | 27/02/2017 | 10/03/2017 |
| 3 | Security & Access Rights | 13/03/2017 | 31/03/2017 |
| 4 | Bonus Features, Archive Users  Compensatory Time if needed | 3/04/2017 | 10/04/2017 |