Project Plan

Digital Imprimatur

Senior Projects

Version 1.1 September 9th, 2022

By

Dominic Rohan
Sam Shadle
John Palladino

1. IEEE Project Plan
    - 1.1. Summary of Project
        - 1.1.1. Purpose/Scope/Objectives
        - 1.1.2. Assumptions/Constraints
        - 1.1.3. Project Deliverables
        - 1.1.4. Schedule/Budget Summary
        - 1.1.5. Evolution of the Plan
    - 1.2. References
    - 1.3. Definitions
    - 1.4. Project organization
        - 1.4.1. External interfaces
        - 1.4.2. Internal structure
        - 1.4.3. Roles/responsibilities
    - 1.5. Managerial process plans
        - 1.5.1. Startup plan
        - 1.5.2. Work plan
        - 1.5.3. Control plan
        - 1.5.4. Risk management plan
        - 1.5.5. Closeout plan
    - 1.6. Technical process plans
        - 1.6.1. Process model
        - 1.6.2. Methods, tools, and techniques
        - 1.6.3. Infrastructure plan
        - 1.6.4. Product acceptance plan
    - 1.7. Supporting process plans
        - 1.7.1. Configuration management plan
        - 1.7.2. Verification and validation plan
        - 1.7.3. Documentation plan
        - 1.7.4. Quality assurance plan
        - 1.7.5. Reviews/audits
        - 1.7.6. Problem resolution plan
        - 1.7.7. Subcontractor management plan
        - 1.7.8. Process improvement plan
    - 1.8. Additional plans
2. Scenarios
    - 2.1. Publisher
    - 2.2. Reviewer
    - 2.3. Verifier

1 Overview

1.1 Summary of Project

1.1.1 Purpose/Scope/Objectives

The Catholic Church's existing Imprimatur process, which has existed for thousands of years, utilizes inefficient physical and paper methods and no digital technology. In modern times, the internet is the most accessible tool used for access books which therefore the church must think about. As a result:

- The review process takes over 9 months per work, and
- There is no way to verify the integrity of a work once it is approved

This product will allow a church authority to manage the Imprimatur process digitally in a time-effective way. Once approved and given a digital Imprimatur, a work cannot be digitally altered without losing the Imprimatur.

Our project will provide a way to collect and store documents for review, automate the review process, allow human in-the-loop intervention, provide a user friendly interface, a secure digital stamp of approval, provide adequate cybersecurity to block hackers or embedded nefarious content, and lock down the final digital version.

1.1.2 Assumptions and Constraints

- For this class resources are not initially required
- Produce viable presentation for user review/comment of proposed new process for Imprimatur by assigned date (November)
- Must work with the existing process of Imprimatur approval (need to know if each diocese a unique process)
- Easy to use for non-technical users

Dependencies

- The Church to provide existing imprimatur process
- Church decision to manage by USCCB, or each Diocese
- Church existing tools

1.1.3 Project Deliverables

- General User: Front end UI for publisher to allow ability for users to submit documents for review
- Diocese User: Interface for verification offices to review submitted documents
- Use tools to automate the review process (e.g., Workflow) by Diocese that provides a report of findings. Record the trail of reviewers and results.
- Allow for manual inspection of the submitted article

- All for the addition of a Digital Imprimatur in the document (and automatically lock down the document), and its appearance in product
- Add function to store the document, and provide sender a copy
- Way for end consumers to verify the Imprimatur and authenticity of document

1.1.4 Schedule/Budget Summary

- Project Plan – Draft: Sep 12 | Final: Sep 19
- Requirements – Draft: Sep 26 | Final: Oct 3
- Architecture – Draft: Oct 10 | Final: Oct 17

1.1.5 Evolution of the Plan

- After each formal submittal, each artifact may evolve given new discoveries

1.2 References

1.2.1 Canon Law, Diocese Process (TBD)

1.3 Definitions

Imprimatur: (Latin: “let it be printed”), in the Roman Catholic church, a permission, required by contemporary canon law and granted by a bishop, for the publication of any work on Scripture or, in general, any writing containing something of peculiar significance to religion, theology, or morality. [reference]

1.4 Project Organization

1.4.1 External structure

Class Weekly scrum meetings with Instructor, Diocese stakeholder(s), USCCB and Project Organizer (PO) (Jon C).

1.4.2 Internal structure

The members of the development team will meet weekly to discuss our work and our responsibilities for the upcoming week. Discuss and commit to roles and responsibilities e.g.

- Record meeting notes, and Instructor, Diocese, USCCB and PO comments.
- Conduct Research and record/share
- Develop system components and review as a team and stakeholders.

1.4.3 Roles and responsibilities

Each role will be primarily, but not entirely responsible for their part of the project.

Examples:

- Record meeting notes
-

Project Manager

1.5 Managerial Process Plan

1.5.1 Project Start-up Plan

1.5.1.1 Estimation Plan

At this moment, the production of this prototype will need no funding from outside organizations, but may need to accommodate resources if/as provided. The majority of the project will be done by our team with potential stakeholder assistance. We will be meeting weekly internally and potentially with varied stakeholders for more information about our clients needs and expectations.

When the team meets weekly, it will collect ideas and continue work on project plans, and how the execution of each role is performing. We will track risks, issues, successes, obstacles, and opportunities. If a risk or issue is encountered, the team will devise mitigation plans to track and work each one. It is crucial that all members are present during all meetings due to the few members of our team.

The team will use Jira Work Management to help plan, assign, track, report, and manage work.

If necessary, the cost or schedule will be reevaluated to fit the needs of the project.

1.5.1.2 Staffing Plan

All members of the team are expected to have sufficient programming abilities or access to resources in order to keep up with the pace of the team. Roles will be assigned to team members and team members are expected to implement their roles within each phase of the project plan. No other staff will be necessary for completion of the project.

ADD APPENDIX IN BACK OF DOCUMENT ROLES, PERSON ASSIGNED, SKILLS

1.5.1.3 Resource Acquisition Plan

For this class resources are not initially required, they will be available through FUS, or personally owned equipment. If there is a need beyond this, it will be provide by the stakeholder community. The Acquisition or need for resources by the stakeholder community will be reevaluated throughout the course of project due to insufficient knowledge of processes.

1.5.2 Work Plan

1.5.2.1 Work Activities

The months ranging from September through December will be used to discover more alleyways for the proposed project. This time will be spent researching and acquiring needed resources (unknown).

Jira Management will be used to keep track of requirements and assignments, made by the team, to help further progression of the project.

Weekly meetings will help to further direct our focus on current parts of the project: Project Plan, Requirements, and Architecture.

We will conduct a bi-weekly review of the plan (sprint) to ensure ability to present accomplished work done.

1.5.2.2 Schedule Allocation

- Project Plan – Draft: Sep 12 | Final: Sep 19
- Requirements – Draft: Sep 26 | Final: Oct 3
- Architecture – Draft: Oct 10 | Final: Oct 17

1.5.3 Control Plan

The Jira tool will enable the management of project membership and will be used to help plan, assign, track, report, and manage work. Work will be assigned to individuals on the team. Work will have due dates, requirements, and expectations (result). Each ticket (assignment) will have a series of steps: available, in progress, in review, in test, and finished.

1.5.3.1 Quality control plan

Class meetings with team leaders will allow presenting, testing, and evaluation of plan/product to ensure the best quality work for the stakeholders.

1.5.4 Risk Management Plan

By conducting weekly meetings and the use of Jira management, an organized and visible representation of the work completed will be readily available to team leaders to prevent low volume of work. Also, the project will keep a rolling, persistent record of risks/issues e.g., spreadsheet in Excel, for tracking purposes.

1.6 Technical Process Plans

1.6.1 Process model

1.6.2 Methods, tools, and techniques

- Blockchain (Etherium) and Industry-standard encryption techniques, including an encryption scheme to "lock down" files once they have been granted an Imprimatur (such as hashing)

Still needed

- Continuous integration software/testing
- Infrastructure/APIs for automated review of documents

1.6.3 Infrastructure plan

- For project development, everyone should be able to use existing hardware and networks; no purchases should be required for development (there is also the existing lab in JSerra)
- For the project code: git/github
- For podcasts: RSS feeds
- Javascript (React with MUI for frontend, Node for backend)
- MySQL for database
- For documenting the project progress and sprints: Jira
- Stretch goal: provide as SaaS → Underlying infrastructure necessary for this
- *(If we do the verification piece in a correct/decentralized way, it shouldn't be necessary for us (or any one place) to host content; the publishers should be able to host all content)*

1.6.4 Product acceptance plan

- Content types: This system should be able to accept text, images, video, and audio files in standard formats (we should define exactly which formats). For security, any files whose types do not match the specified formats should be automatically rejected
- Process: The approval process of a piece of content should conform to the existing process in any given diocese. To test this, we should set it up using the requirements of at least 2-3 different dioceses (with up to 5 as a stretch goal) and make sure everything still works
    - Pre-screening: depending on the requirements specified by each diocese, the system should be capable of automatically flagging certain suspicious words, phrases, or images. To test this, we should submit content which contains these forbidden words/images, and make sure the system flags each one appropriately
- Authenticity: It should be impossible to covertly tamper with a piece of content without it losing its verification of Imprimatur. To test this, we should make every conceivable type of modification to a piece of content, and make sure it loses its verification status each time

1.7 Supporting Process Plan

1.7.1 Configuration Management Plan

Our software will be used in dioceses across the United States, all of which have different rules for the process of the imprimatur.
The software must be able to manage configurations for each diocese separately and at the same time.
Configurations will be created by reviewers in each diocese and can be updated at any time.
Configurations written for each diocese must be able to express the variation in the review process, and the process to configure one must be intuitive to someone familiar to the imprimatur process.
Writing of configuration will be handled at the diocese level, and the project team will manage what can be expressed in the configuration.

1.7.2 Verification and Validation Plan

As software development progresses, we will write unit tests to validate every component we have written works. Every time a feature is worked on, the validation team should write tests for it.

A continuous integration system (CI/CD) will be used to ensure the tests aren't broken. We will verify the software by meeting often and getting representative users to review it. First, we will create prototypes to get a general idea of where the software is going and go over them with the client.
Once full development is underway, we will create a backlog of capabilities allocated to sprints, create a milestone every once in a while and perform a verification process on it. The verification process will include getting the client's approval, testing by members of the team, and by representative users.

1.7.3 Documentation Plan

The digital imprimatur will be heavily documented internally and externally. We will provide operator documentation in the form of training videos distributed along with the project. These training videos will be accessible through the interface. In addition, each page will have a help button which shows how to do the most common tasks.
We will also provide systems documentation for the IT departments. The systems documentation will be distributed as a PDF given along with the software, and will contain instructions on how to install, maintain and secure our software.
Finally, we will maintain developer documentation. We will maintain a list of software requirements, a high-level architecture of the software, and test plans. All code will be commented, and each function will be commented with a description of what it does. We will generate a documentation site from the comments.

1.7.4 Quality Assurance Plan

The primary goal of this project is to move the slow, physical process of review for the imprimatur over into the digital world. To accurately recreate this process, we must be meticulous to keep our digital process the same as the physical process. To ensure that we have done this, once we have decided on how the review process works, we will have it reviewed by a canon lawyer, or a representative of the diocesan offices. The information we gather during this meeting will be used to create another digital model, which will undergo review again until it is found to be satisfactory..

1.7.5 Reviews and Audit Plan

The project will be reviewed at every sprint and milestone. The review process will contain live testing by the client and some representative users. The milestone version of the software will be deployed to a testing environment where it can be accessed by testers, and testers will get a chance to give their thoughts on the software. Members of the development team will look back over their own and their teammates' code, and the project organizer will review the code.

1.7.6 Risk and Issue Resolution Plan

Problems will be reported using the Jira bug tracking software. Problems will be analyzed and prioritized during scrum meetings and assigned to a problem owner. The problem owner will be responsible for fixing it, creating a report of how they fixed it, and reporting on how the process of problem resolution went.

1.7.7 Subcontractor Management Plan

We don’t intend to use subcontractors in this project.

1.7.8 Process Improvement Plan

The project should be assessed every sprint and milestone, or after a major breaking change. During the assessment, an analysis of what has changed in the project will be conducted. If

the changes made are significant enough to warrant a change to the plans, changes will be made and recorded.

2 Scenarios

2.1 Publisher

Scenario: A publisher wishes to get the bishops' seal of approval for their new book about the faith in order to verify its teachings.

Requirements:

- The publisher is able to submit the book in their local diocese for review
- The publisher can view the status of the in-progress review, as it moves from pending, to under review, to imprimatur
- The publisher can withdraw their request for an imprimatur
- The publisher can control which dioceses the book is submitted in

2.2 Reviewer

Scenario: A reviewer will have the ability to run a document through a series of processors which will edit the document limiting the excessive time spent on the review process.

Requirements:

-

2.3 Verifier

Scenario: A user wishes to listen to the "Bible in a Year" podcast, published by Ascension Presents. The podcast episode in question has already been granted a digital Imprimatur.

Requirements:

- The user is able to easily and reliably verify that this episode has been granted a digital Imprimatur, regardless of the platform or service they use to access it (likely via a link in the description)
- If the episode has been altered in any way, the content will not be served and the episode will not play

Thoughts on how to accomplish this:

For podcasts:

- Regardless of service, each podcast episode has a GUID ("Globally Unique Identifier"). By default, this ID is a string that is identical to the URL from which the podcast episode is served
    - If a bad actor changes the URL of the podcast to their own website, the GUID would also change, and it would appear as a separate podcast
    - If the episode has been tampered with, the website from which it is served will not serve the content, and instead serve an audio error message
- The main consideration is the "handshake" between the podcast player and the service website the podcast; I'm having trouble thinking of how this is possible

https://cours.etsmtl.ca/log792/private/restreint/IEEE_1058_Project_Management_Plan.pdf