PETPAL

SOFTWARE CONFIGURATION PLAN

Version <1.0>
10/25/2024

This Template is the property of Cyrille Michaud
License terms : see http://blog.cm-dm.com/post/2011/11/04/License

TABLE OF CONTENTS

This Template is the property of Cyrille Michaud
License terms : see http://blog.cm-dm.com/post/2011/11/04/License

1 Identification

1.1 Document overview

This document contains the software configuration management plan of software Petpal.

2 Organization

The software configuration is managed by members of the project, with specific tools. Responsibilities are shared between

- Release Manager (Najah Ismail),
- Project Manager (Mishra Apurva),
- Lead Developer (Gambhir Dhruv).

2.1 Activities and responsibilities

Activities when setting up the project	**Person responsible**
Identify the configuration items	Release Engineer
Install the bug repository tool and set up the database	RM , Back End Developer
Install the software configuration repository tool and set up the database	Lead Developer
Manage and structure the reference space	Release Engineer, Lead Developer
Define the configuration processes	Release Engineer, Lead Developer

Activities during the project lifecycle	**Person responsible**
Export components for modification, test or delivery	Lead Developer
Create version, write version delivery document	Release Engineer
Approve reference configurations	Project manager
Verify version to be delivered and authorise deliveries	Project manager
Backup spaces	Release Engineer, Lead Developer
Do configuration audits	Quality Manager
Inspect configuration records	Quality Manager
Archive reference version	Release Engineer

Management activities	**Person responsible**
Manage versions and archives	Release Engineer
Manage configuration records	Release Engineer
Produce reports and statistics	Release Engineer
Manage reference space and its access control list	Release Engineer
Manage spaces backup and archive media	Release Engineer
Manage quality reports	Quality Manager

2.1.1 Decisions process and responsibilities

Responsibilities during reviews, audits and approvals are listed below:

At the end of an activity of the project

This Template is the property of Cyrille Michaud
License terms : see http://blog.cm-dm.com/post/2011/11/04/License

Activities	Person Responsible
Do a configuration freeze	Release Engineer, Lead Developer
Present a configuration state of the components impacted by the activity	Release Engineer
Present a documentation state of the components impacted by the activity	Release Engineer

During a configuration management process audit:

Activities	Person Responsible
Do the configuration management process audit	Project Manager
Present the records of the configuration management process	Release Engineer
Present the quality records of the configuration management process	Quality Manager
Present the records of the documentation management process	Release Engineer

3 Configuration identification

3.1 Identification rules

3.1.1 Identification rules of configuration items

3.1.1.1 Identification of a configuration item

A version number is tagged to any delivery of a configuration item. This number shall be incremented before each delivery, especially if modifications are made to the product.

3.1.1.2 Version number of a configuration item

The version number will follow the following scheme: XXX_Vm.n

Where:

- XXX: A number identifier for the configuration item.
- V: Denotes the “version”
- m. Major version number, pertaining to releases which change major functionalities.
- n. Minor version number, incremented for non-breaking software changes.

When preparing a new delivery, the version number should be updated as follows:

- Major version (m): incremented when there are substantial changes or enhancements.
- Minor version (n): incremented when small, non-breaking changes or fixes are introduced.

3.1.2 Identification rules of documents

3.1.2.1 Description of documents identifiers

The identification of documents is described below:
<document type>_<document number>.<revision index>
where:

This Template is the property of Cyrille Michaud
License terms : see http://blog.cm-dm.com/post/2011/11/04/License

" Document type " is described by the title of the document. For example: Project_Plan, Risk_Management_Plan
" Document number " is a incremental number, with a separate list for each document type,
" Revision index " designates the approved iteration of the document.
Example: Project_Plan_1.1

3.1.2.2 Definition and evolution of the revision index

The attribution of a revision index is a prerequisite to any delivery of a document or file. This index shall be incremented before the diffusion of a modified document.
The definition rules of a revision index are the following:

Example of the revision index:

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason

3.2 Reference configuration identification

Each reference configuration is defined by:

- An identifier,
- Its content listed in the corresponding Version Delivery Description document,
- The acceptance or validation reviews associated with the building of the reference configuration.

A reference configuration is established for each design review and each test review of the project.

3.3 Configuration Baseline Management

- functional baseline (FBL);
  - Authentication must ensure secure registration and login for all user types
  - Pet adoption feature is facilitated by search and register interest for User. Agencies can create and manage pet profiles.
  - Pet sitting feature is facilitated by enabling creation of requests as well as search and registration for sitting jobs.
  - Pet sitting feature is facilitated by enabling creation of Events as well as search and register for existing events.

  It will be defined during the requirements gathering phase, with stakeholder input, and controlled through change management processes to ensure that all functionalities meet user needs.

- allocated baseline (ABL):
  - Front-end : Built using Next.js, ensuring responsive design with server-side rendering (SSR) for faster load times
  - Back-end : Developed with Flask as a lightweight Python framework for handling API requests.
  - Database: PostgreSQL as the primary database, utilizing SQLAlchemy for Object-Relational Mapping (ORM)

This will be established during the design phase and controlled through design reviews and configuration management to ensure alignment with the FBL.

- product baseline (PBL)
  - Fully operational and completed features for Adoption, Events and Sitting.
  - Security implemented through authentication for all users.
  - Minimally the following documentation must be complete with the latest software version:
    - Project Proposal
    - Configuration Management Log
    - Risk Management

Baseline documents are maintained and updated in shared cloud resources (OneDrive and Google Drive) accessible to all the members of the development team.

4 Configuration control

4.1 Change Management

- Changes requests are emitted from by any member of the team according to the problem resolution process,
- When a change request is accepted by the project manager/product manager, a branch is created in the Version Control System (git).
- The branch is identified by naming it in a descriptive manner of the feature being added. This name is agreed upon by all members of the team.

4.2 Interface Management

3rd party interfaces (Firebase Authentication) are checked for any system disruption or change. A management document will be produced in an ad hoc manner if necessary.

5 Configuration support activities

5.1 Configuration Status Accounting

Configuration Status Accounting (CSA) is the process to record, store, maintain and report the status of configuration items during the software lifecycle. All software and related documentation are tracked throughout the software life.

5.1.1 Evolutions traceability

Modifications to various types of configuration items will be tracked as follows:

- Documents: The modification sheet will include a summary of the changes, with specific paragraphs or sections marked for easy identification.
- Source Files: The Git commit messages will record modifications at the file level, describing what was changed and why.

The modification sheet describes the modifications done to the components with enough precision to identify the modified parts.

5.1.2 Setting up Configuration status

The Release Engineer will establish the configuration status by labelling each version with:

This Template is the property of Cyrille Michaud
License terms : see http://blog.cm-dm.com/post/2011/11/04/License

- The version number,
- The creation date,
- The relevant information updated in the Change Management Log

5.1.3 Configuration status diffusion

Both the Release Engineer and QA Manager will collaborate to update the Change Request document for each new version.

5.1.4 Configuration status records storage

The records are stored in a configuration folder, which contains:

- The requests sorted by record number,
- The software documents,
- The configuration states are sorted chronologically.

5.2 Configuration audits

Following configuration audits will be conducted to ensure compliance with this plan:

- baseline audit
- functional configuration audit
- software configuration audit.

These will be reviewed during the team meetings.

5.3 Reviews

Technical reviews will assess progress and confirm that configuration items are up-to-date. The Release Engineer will play a central role in ensuring that baselines and branches are properly maintained. These are also conducted during the bi-weekly team meetings.

5.4 Configuration management plan maintenance

The Release Engineer is responsible for maintaining this plan. Updates will occur periodically, or whenever there are major changes to the project. Responsibilities include ensuring the plan is kept current throughout the project lifecycle.

This Template is the property of Cyrille Michaud
License terms : see http://blog.cm-dm.com/post/2011/11/04/License