

Configuration Management Plan



UnisaEAT

Riferimento	
Versione	1.0
Data	06/11/2021
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Amideo Salvatore e Vidoni Alice
Approvato da	

Sommario

Revision History	2
Team Composition.....	3
1. Introduzione	4
1.1 Ambito.....	4
1.2 Scopo del documento	4
1.3 Riferimenti	4
2. Management	5
2.1 Fasi del progetto	5
2.2 Organizzazione	5
2.3 Ruoli e Responsabilità	6
3. Attività	6
3.1 Configuration Identification	7
3.2 Configuration Item.....	7
3.3 Configuration Control	8
3.4 Configuration Version Release (CVR)	9
3.5 Configuration Status Accounting	9
3.6 Configuration Audit	9

Revision History

Data	Versione	Descrizione	Autori
04/11/2021	0.1	Prima stesura	Salvatore Amideo Alice Vidoni
06/11/2021	1.0	Revisione Finale	Salvatore Amideo Alice Vidoni

Team Composition

Ruolo	Nome	Posizione	Contatti
Sponsor	Filomena Ferrucci	Sponsor	fferrucci@unisa.it
Project Manager	Alice Vidoni	Project Manager	a.vidoni@studenti.unisa.it
Project Manager	Salvatore Amideo	Project Manager	s.amideo@studenti.unisa.it
Team Member	Alessandro Cavaliere	Team Member	a.cavaliere41@studenti.unisa.it
Team Member	Alessio Salzano	Team Member	a.salzano32@studenti.unisa.it
Team Member	Carmine Citro	Team Member	c.citro23@studenti.unisa.it
Team Member	Claudio Buono	Team Member	c.buono20@studenti.unisa.it
Team Member	Gerardo Sessa	Team Member	g.sessa56@studenti.unisa.it
Team Member	Maria Rosaria Salzano	Team Member	m.giudice12@studenti.unisa.it
Team Member	Nicola Cappello	Team Member	n.cappello@studenti.unisa.it

Configuration Management Plan (CMP) del Progetto UnisaEAT

1. Introduzione

1.1 Ambito

Il Sistema **UnisaEAT**, si pone come obiettivo la semplificazione e la velocizzazione delle funzionalità del servizio mensa UNISA, rendendole più agevoli sia per chi ci lavora sia per chi ne usufruisce. I nostri obiettivi riguardano la realizzazione di un sistema che permetta di avere una massima automazione per quanto riguarda l'intera gestione del tesserino mensa ed automatizzare le ordinazioni dei pasti. Si vuole velocizzare le comunicazioni tra Cliente e personale ADISU, ottimizzare la metodologia d'acquisto dei pasti, facilitare le mansioni dell'operatore Mensa nella distribuzione dei pasti e agevolare le funzioni gestionali dell'intero servizio.

1.2 Scopo del documento

Per consentire il corretto progresso del progetto e la sua riuscita risulta necessario produrre un Configuration Management Plan (CMP). In tale documento saranno trattati ruoli nell'ambito della configuration mangament, configuration items e modifiche durante lo sviluppo del progetto.

1.3 Riferimenti

- Kathy Schwalbe, "Information Technology Project Management", International Edition 7E, Cengage Learning, 2014;
- Bernd Bruegge, Allen H. Dutoit, "Object-Oriented Software Engineering Using UML, Patterns and Java", Third Ed., Pearson, 2010;
- Sommerville, "Software Engineering", Addison Wesley;
- PMBOK® Guide and Software Extention to the PMBOK® Guide, Fifth Ed., Project Management Institute, 2013;
- Documentazione di Progetto.

2. Management

2.1 Fasi del progetto

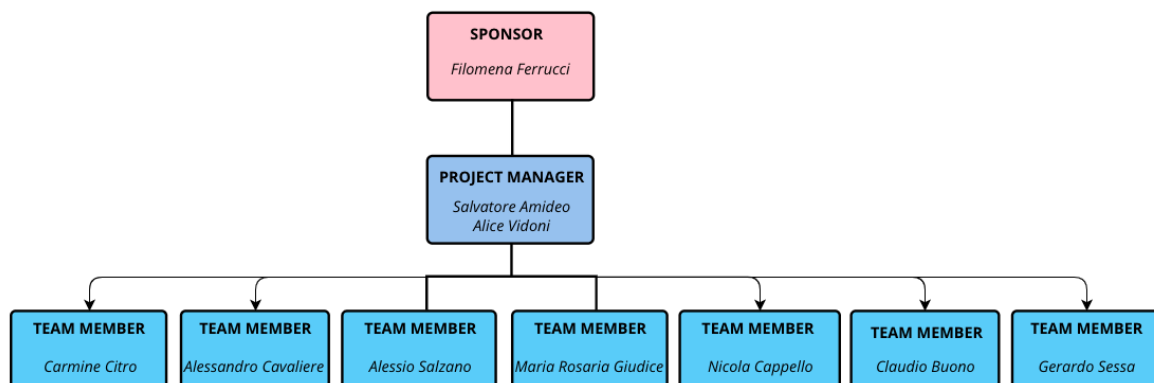
Sono state identificate le seguenti principali fasi di progetto:

1. Avvio del progetto
2. Requirements Elicitation and Analysis
3. System Design
4. System and Integration Test Design
5. Object Design
6. Implementazione
7. Testing
8. Rilascio

Oltre le precedenti 8 fasi, per tutta la durata del progetto verranno eseguite continue attività di Training atte ad aiutare i team members nei loro lavori.

2.2 Organizzazione

Di seguito è mostrato un diagramma dell'organizzazione.



2.3 Ruoli e Responsabilità

I seguenti ruoli e responsabilità riguardano il Configuration Management Plan per il progetto UnisaEAT.

Project Manager

Il Project Manager è la figura responsabile della gestione di tutti gli aspetti manageriali del progetto. Il Project Manager è responsabile per:

- Identificazione dei Configuration Items
- Accettazione di change requests.
- Assegnazione delle responsabilità delle change requests.

Revision Leader

Il Lead Engineers è un membro del team di sviluppo che ha maggiori responsabilità nell'ambito della configuration management rispetto ai suoi colleghi per delle determinate task.

Il Lead Engineers è responsabile per:

- Eseguire revisioni degli artefatti prodotti
- Comunica al PM risultati riguardanti le revisioni effettuate
- Comunica al PM eventuali necessità di cambiamenti negli artefatti prodotti

È stato deciso che questo ruolo verrà svolto a rotazione da tutti i membri del team.

Developer

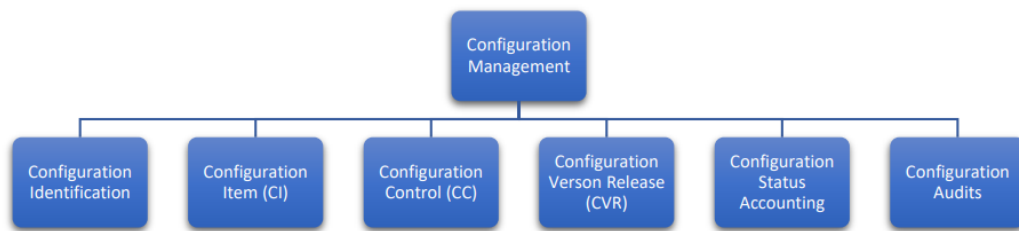
Ogni membro del team riveste il ruolo di engineer all'interno del progetto.

Lo Engineer è responsabile per:

- Eseguire revisioni programmate assegnategli dal PM
- Compilare i documenti di revisione
- Implementare le change requests per gli artefatti prodotti

3. Attività

Il Configuration Management consta delle seguenti attività:



3.1 Configuration Identification

Una volta identificati i Configuration Item, il Project Manager, seguendo le direttive del Top Manager, assegnerà un nome allo item e lo inserirà nel rispettivo CMDB in uno stato iniziale. Subito dopo, l'item sarà assegnato al team di sviluppo che dovrà lavorarci e tenere traccia delle modifiche attraverso un log con identificativo incrementale, spesso situato nell'artefatto stesso.

3.2 Configuration Item

Ricadono nella categoria di Configuration Item:

- Documenti per la gestione e l'esecuzione del progetto.
- Documenti riguardanti lo sviluppo del sistema.
- Documenti di carattere tecnico del sistema.
- Applicativo software con relativa documentazione.
- Altri documenti a discrezione del PM.

Ogni Configuration Item è caratterizzato da:

- Un identificativo univoco solitamente della forma "2021_[Acronimo Documento]_C3_Amideo-Vidoni_V[x.y]".
- Un numero di versione, della forma "x.y" dove x è usato per indicare la consegna o il raggiungimento di una milestones, mentre y per indicare modifiche di minore entità.

Al momento della sua individuazione e inserimento nel CMDB, allo item viene assegnata la versione 0.1 indicante lo stato di iniziazione. Successive modifiche vanno a incrementare solo la y della versione. Alla prima release del documento o al raggiungimento di una milestones ad esso dedicata, la versione diventa 1.0. Da tale momento in poi ogni modifica allo item deve essere approvata da PM e sottoposta secondo protocollo e porterà all'incremento della sola y. La x verrà nuovamente incrementata solo in caso di nuove consegne, nuove milestones, o modifiche di natura particolarmente importanti, a discrezione del PM.

Configuration Management Database (CMDB)

Nell'ambito del progetto UnisaEAT sono stati individuati due diversi CMDB:

- One Drive: per la gestione dei documenti riguardanti il design dell'applicativo (RAD, SDD, ODD, e altri).
- GitHub: per la gestione del codice sorgente di UnisaEAT.

Entrambi i sistemi offrono nativamente funzioni per il versioning degli artefatti, alle quali sono state aggiunte ulteriori protocolli per irrobustire il tutto. Nello specifico, nel caso dei documenti su One Drive, ogni artefatto ha internamente una tabella "Revision History" le quali entry associano ad una specifica versione del documento una descrizione della modifica implementata. Ancora, ogni rilascio di una versione con modifiche importanti (che quindi vanno a incrementare il primo numero della versione) prevedono la creazione di una nuova copia del documento separata dalla precedente versione. Nel caso di GitHub, invece, ci si affida completamente al sistema basato su commit del sito stesso, andando però a evidenziare in maniera chiara un commit di rilascio corrispondente alle versioni con l'incremento del primo numero.

3.3 Configuration Control

Una volta che un Configuration Item rientra nella baseline, tutte le modifiche ad esso fatte devono seguire un preciso protocollo. Chiunque facente parte del team può sottoporre una change request al Revision Leader che ha il compito di fare una valutazione preliminare. Se il Revision Leader approva la change request, essa passa al Project Manager che la valuta e decide se approvarla o rifiutarla. Se approvata, l'implementazione della modifica viene assegnata ad uno o più developer, eventualmente lo stesso che l'ha proposta, che, dopo averla implementata, si assicurerà della sua corretta propagazione e consistenza. Nel caso di modifiche al codice, l'implementatore dovrà assicurare la corretta build e pass dei test dopo l'implementazione della change request, eventualmente avvalendosi dei sistemi di continuous integration offerti da GitHub Action. Sarà poi compito del PM accettare la modifica e integrarla nel progetto effettivo. Per quanto riguarda gli accessi, l'intero team di sviluppo avrà privilegi completi sulla cartella One Drive, intesi come scrittura e lettura degli artefatti. Nel caso della repository GitHub invece, i team members dovranno lavorare usando il sistema di branch offerto dalla piattaforma e sottoporre le proprie modifiche mediante pull request sui i branch master (per le release) e develop (per lo sviluppo). Solo il PM, una volta assicuratosi della qualità delle modifiche implementate, potrà accettare la richiesta e integrare i commit di modifica nel progetto effettivo.

3.4 Configuration Version Release (CVR)

L'attività di management delle release è svolta nel momento in cui si verificano condizioni che determinano il rilascio di una nuova release. Esse sono:

- La risoluzione di uno o più bug porta ad un incremento della .y all'interno della versione del codice sorgente;
- La risoluzione o la modifica di piccole parti della documentazione dovute ad inconsistenze rilevate porta all'incremento della .y all'interno della versione del documento;
- La risoluzione di uno o più bug di notevole entità porta all'aggiornamento della X all'interno del codice sorgente;
- Il completamento di un documento con l'aggiunta e la modifica di un numero ingente di sezioni porta all'aumento della X.

3.5 Configuration Status Accounting

Durante il tempo che intercorre tra due consegne o milestones, i CI sono conservati nella cartella One Drive o su GitHub, nelle quali avvengono le principali attività di modifica. Lo accounting dello stato della configurazione avverrà ogni milestones importante o release (consegna) degli artefatti prodotti. In tale situazione, il PM dovrà creare una cartella contenente tutte le versioni più recenti, e approvate, degli artefatti richiesti, eventualmente aggiungendo precedenti versioni per motivi didattici. All'interno di tale cartella, sarà compreso un file "README" con lo scopo di informare chi visiona di informazioni utili alla comprensione degli artefatti. Nel caso del codice sorgente, sarà effettuato un commit univocamente identificato con un numero di rilascio indicante il raggiungimento della milestones o deliverables. Sarà inoltre prodotto un eseguibile dell'applicativo contenuto anch'esso nella cartella con gli altri artefatti.

3.6 Configuration Audit

Al raggiungimento di una milestone, o poco prima una consegna, saranno effettuati dei lavori di revisione da parte di tutto il team di sviluppo su ogni CI facente parte della baseline. Dopo tali lavori, il PM visionerà personalmente gli items assicurando:

- La corretta numerazione delle versioni
- La consistenza delle modifiche tra items collegati
- La qualità delle descrizioni delle modifiche implementate
- La presenza di tutti gli items



- La corretta organizzazione dei CMDBs

In caso di necessità, il PM potrà richiedere il rollback di alcune modifiche. Tale rollback avverrà avvalendosi dei sistemi integrati in One Drive e GitHub