

Norme di progetto

2023-11-13 — v0.0.3



overture.unipd@gmail.com

Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo Overture
Responsabile	Alex Vedovato
Redattori	Riccardo Fabbian Francesco Furno Alex Vedovato
Verificatori	Eleonora Amadori

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
0.0.3	2023-11-13	Francesco Furno	Eleonora Amadori	Aggiunta della sezione 'Documentazione'
0.0.2	2023-11-11	Riccardo Fabbian	Eleonora Amadori	Aggiunta delle sezioni 'Verifica' e 'Validazione'
0.0.1	2023-11-11	Alex Vedovato	Eleonora Amadori	Struttura iniziale del documento ed introduzione

Indice

1) Introduzione	4
1.1) Scopo del documento	4
1.2) Scopo del progetto	4
1.3) Glossario	4
1.4) Riferimenti	4
1.4.1) Riferimenti normativi	4
1.4.2) Riferimenti informativi	4
2) Processi primari	5
3) Processi di supporto	5
3.1) Documentazione	5
3.1.1) Scopo	5
3.1.2) Descrizione	5
3.1.3) Aspettative	5
3.1.4) Ciclo di vita	5
3.1.5) Sistema di composizione tipografica	5
3.1.6) Struttura dei documenti	6
3.1.6.1) Intestazione	6
3.1.6.2) Registro delle modifiche	6
3.1.6.3) Indice	6
3.1.6.4) Corpo del documento	6
3.1.6.5) Corpo del verbale	6
3.1.6.6) Documenti del progetto	7
3.1.7) Regole stilistiche	7
3.1.7.1) Nomi assegnati ai file	7
3.1.7.2) Stile del testo	7
3.1.7.3) Formato delle date	8
3.1.8) Strumenti	8
3.2) Verifica	8
3.2.1) Scopo ed aspettative	8
3.2.2) Descrizione	8
3.2.3) Analisi statica	9
3.2.4) Analisi dinamica	9
3.3) Validazione	9
3.3.1) Scopo ed aspettative	9
3.3.2) Descrizione	10
3.4) Gestione della configurazione	10
3.5) Gestione della qualità	10
4) Processi organizzativi	10

1) Introduzione

1.1) Scopo del documento

Questo documento ha lo scopo di descrivere le regole relative al Way of Working adottato da parte del gruppo per lo svolgimento del progetto didattico. In esso, dunque, appaiono tutte le best practices da seguire per ciascun processo e correlate attività che lo compongono, seguendo nel fare ciò la struttura definita dallo standard ISO/IEC 12207:1995.

Per la stesura è stato intrapreso un approccio di tipo incrementale, ovvero che prevede una realizzazione in più passi con aggiunte successive ad un impianto base. Di conseguenza ogni aggiornamento avverrà in funzione delle decisioni prese dal gruppo durante lo svolgimento del progetto stesso.

I membri del gruppo si impegnano a visionare regolarmente questo documento e a rispettare con disciplina le regole definite in esso, per fare in modo di essere professionali, coerenti, sistematici ed uniformi nello svolgere il lavoro necessario.

1.2) Scopo del progetto

Il progetto richiede lo sviluppo e la valutazione di un'estensione dello standard in Carbonio: questa deve mantenere la compatibilità con i client attualmente supportati e permettere l'espansione delle funzionalità per i client di nuova generazione.

L'obiettivo che si vuole raggiungere è essere in grado di valutare se conviene investire tempo e denaro per integrare il protocollo JMAP su Carbonio. Una volta completato, infatti, il servizio sarà impiegato internamente presso Zextras insieme agli stress test sviluppati dal gruppo per valutare le prestazioni, la manutenibilità e la completezza del protocollo in questione. Tale valutazione sarà effettuata in confronto con i protocolli attualmente sviluppati in Carbonio e permetterà di rispondere effettivamente alla domanda posta dal capitolato.

1.3) Glossario

Per evitare ambiguità o incomprensioni riguardanti la terminologia usata nel documento, è stato deciso di adottare un glossario in cui vengono riportate le varie definizioni. In questa maniera in esso verranno posti tutti i termini specifici del dominio d'uso con relativi significati.

La presenza di un termine all'interno del glossario viene indicata applicando *questo stile*.

1.4) Riferimenti

1.4.1) Riferimenti normativi

- Capitolato d'appalto C8: JMAP, il nuovo protocollo standard per la comunicazione email
<https://www.math.unipd.it/~tullio/IS-1/2023/Progetto/C8.pdf>
- Standard ISO/IEC 12207:1995:
https://www.math.unipd.it/~tullio/IS-1/2009/Approfondimenti/ISO_12207-1995.pdf

1.4.2) Riferimenti informativi

- I processi di ciclo di vita del software
<https://www.math.unipd.it/~tullio/IS-1/2023/Dispense/T2.pdf>

2) Processi primari

3) Processi di supporto

3.1) Documentazione

3.1.1) Scopo

Il processo di documentazione serve a tenere traccia di tutti processi e attività relativi al ciclo di vita del software, riportando le decisioni adottate e le norme attuate dal gruppo durante lo svolgimento del progetto. Le norme stabilite all'interno di questo documento verranno rispettate da tutti i membri del gruppo *Overture*.

3.1.2) Descrizione

La documentazione software traccia il lavoro svolto e le decisioni prese. Questa sezione del documento si occupa di tutte le norme adottate dal gruppo relative alla documentazione.

3.1.3) Aspettative

- Definire delle regole chiare e concise per la stesura di tutti i documenti relativi al progetto
- Creare template per tutti i tipi di documenti per garantire omogeneità

3.1.4) Ciclo di vita

Il ciclo di vita di un documento è composto da sette fasi:

1. Creazione o adattamento del template: la prima fase prevede la creazione o l'adattamento di un template per il documento corrente. Il template contiene la struttura e la formattazione del documento, nonché le informazioni di base, come titolo, autore e data.
2. Pianificazione e assegnazione delle sezioni: nella seconda fase le sezioni del documento vengono pianificate e assegnate ai Redattori incaricati. I Redattori sono responsabili della stesura delle proprie sezioni in conformità con le norme di progetto.
3. Raccolta dei contenuti e creazione della prima bozza: nella terza fase i Redattori raccolgono i contenuti da discutere e creano una prima bozza del documento. La bozza viene utilizzata come punto di partenza per la discussione e la revisione.
4. Stesura effettiva del documento: la quarta fase vede i Redattori redigere le proprie sezioni in conformità con il modello e le norme di progetto.
5. Controllo dei contenuti: nella quinta fase i Redattori verificano che il contenuto delle proprie sezioni sia conforme alle norme di progetto e non contenga errori di compilazione.
6. Revisione: la sesta fase prevede che un Verificatore incaricato revisioni il documento per assicurarsi che le modifiche apportate siano corrette.
7. Approvazione e rilascio: nell'ultima fase il documento viene approvato da un Responsabile e rilasciato in versione finale.

3.1.5) Sistema di composizione tipografica

Per la composizione tipografica dei documenti, abbiamo deciso di utilizzare Typst, al posto del noto LaTeX. Offre diversi vantaggi rispetto a LaTeX:

- semplicità di utilizzo (simile a Markdown)
- programmabilità reale invece di un sistema di macro
- compilazione pressochè immediata.

L'utilizzo di Typst semplifica la creazione e la manutenzione dei documenti, liberando i redattori dalla responsabilità della visualizzazione grafica e garantendo una certa coerenza nella documentazione del

progetto.

Il template sviluppato ed utilizzato è presente nella repository docs, nella cartella templates.

3.1.6) Struttura dei documenti

Ogni documento prodotto viene organizzato nelle seguenti sezioni:

3.1.6.1) Intestazione

La prima pagina funge da intestazione del documento e presenta gli elementi di seguito:

- **Nome del documento**
- **Data:** la data in cui è stata approvata l'ultima versione del documento
- **Versione:** la versione corrente del documento
- **Logo del gruppo:** presente nel percorso `imgs/group_logo.png`
- **Email:** overture.unipd@gmail.com
- **Destinatari:** a chi è il rivolto il documento
- **Responsabile:** chi ha approvato il documento
- **Redattori:** incaricati della stesura del documento
- **Verificatori:** incaricati della verifica del documento

3.1.6.2) Registro delle modifiche

La seconda pagina è dedicata al registro delle modifiche. Le informazioni sono organizzate in una tabella e permettono di tenere traccia dei cambiamenti subiti dal documento. La tabella riporta i seguenti dati:

- **Versione:** il numero di versione del documento
- **Data:** data di approvazione del documento
- **Autori:** chi ha effettuato le modifiche
- **Verificatori:** chi ha approvato le modifiche
- **Dettaglio:** una breve descrizione

3.1.6.3) Indice

Nella terza pagina è presente l'indice che elenca le sezioni contenute nel documento.

3.1.6.4) Corpo del documento

Il contenuto del documento è suddiviso in capitoli, ognuno dei quali è formato da più sezioni ed eventuali sottosezioni.

3.1.6.5) Corpo del verbale

Il contenuto del verbale è suddiviso nelle seguenti sezioni:

- **Informazioni sulla riunione:**
 - **Luogo:** può essere il luogo fisico dove si è tenuto l'incontro oppure il nome della piattaforma online utilizzata
 - **Ora di inizio**
 - **Ora di fine**
 - **Partecipanti:** i nomi dei componenti del gruppo che hanno partecipato alla riunione
 - **Partecipanti esterni:** i nomi di eventuali partecipanti esterni
- **Ordine del giorno:** un elenco di ciò che verrà discusso durante la riunione
- **Sintesi dell'incontro:** contiene un breve riassunto delle discussioni e dei temi affrontati durante l'incontro.
- **Decisioni prese:** sezione che elenca in forma testuale le decisioni prese durante l'incontro. Alcune di queste potrebbero risultare in "attività individuate"

- **Attività individuate:** illustrazione dettagliata delle attività assegnate ai diversi membri del gruppo a conclusione dell'incontro. Queste informazioni, inserite in un'apposita tabella, riportano:
 - **ID:** collegamento alla relativa issue su GitHub
 - **Dettaglio:** breve spiegazione dell'attività
 - **Assegnatari:** i nomi degli incaricati a svolgere l'attività

3.1.6.6) Documenti del progetto

Verranno prodotti i seguenti documenti:

- Norme di Progetto
- Piano di Progetto
- Piano di Qualifica
- Analisi dei requisiti
- Glossario
- Verbali Interni
- Verbali Esterni

3.1.7) Regole stilistiche

3.1.7.1) Nomi assegnati ai file

I documenti PDF presenti nella repository docs, rispettano le seguenti regole per la nominazione dei file:

- minuscolo per i nomi, tranne che per i verbali (marcati VI e VE, rispettivamente per interni ed esterni)
- spaziatura fra le parole sostituita da un underscore
- per i verbali, la data dell'incontro è presente nel nome
- versione del documento alla fine del nome del file

I nomi dei documenti presenti nel progetto saranno quindi del tipo:

- **Norme di Progetto:** norme_di_progetto_vX.X.X
- **Piano di Progetto:** piano_di_progetto_vX.X.X
- **Piano di Qualifica:** piano_di_qualifica_vX.X.X
- **Analisi dei Requisiti:** analisi_dei_requisiti_vX.X.X
- **Glossario:** glossario_vX.X.X
- **Verbali Interni:** VI_YYYY_MM_DD_vX.X.X
- **Verbali Esterni:** VE_YYYY_MM_DD_vX.X.X

Si noti che i sorgenti .typ non includono la versione nel nome, ma è aggiunta ai PDF dopo la compilazione.

Usare lo stesso nome per i documenti consente di utilizzare Git in modo appropriato: tracciare i cambiamenti di file di testo, con relativa facilità di utilizzo della funzione di "diff".

3.1.7.2) Stile del testo

Nei documenti, esclusi i verbali, verranno utilizzati:

- il *corsivo* per:
 - il nome del gruppo (*Overture*)
 - il nome dell'azienda proponente (*Zextras*)
- il **grassetto** per:
 - le voci degli elenchi puntati
 - parole importanti
- un font monospace per:

- i nomi dei documenti
- i nomi dei file
- i nomi delle cartelle o repository
- esempi di codice
- il sottolineato per:
 - i link
 - l'indirizzo email
- le lettere maiuscole per:
 - le iniziali dei nomi
 - gli acronimi
 - le iniziali dei ruoli svolti dai componenti del gruppo

3.1.7.3) Formato delle date

Viene adottato lo standard internazionale **ISO 8601**, nella forma YYYY-MM-DD, indicante rispettivamente

- YYYY: l'anno con 4 cifre
- MM: il mese con 2 cifre
- DD: il giorno con 2 cifre

3.1.8) Strumenti

I seguenti strumenti sono stati scelti dal gruppo per la realizzazione della documentazione:

- **Typst**: linguaggio per la stesura dei documenti, tramite [typst.app](#)
- **GitHub**: servizio di hosting di repository

3.2) Verifica

3.2.1) Scopo ed aspettative

La verifica nel ciclo di vita del software è un processo continuo che inizia dalla fase iniziale di progettazione e si estende fino alla manutenzione successiva. Questo elemento cruciale mira a garantire la conformità tra gli output del software (codice sorgente, documentazione, test e così via) e le relative aspettative, fondandosi su criteri come coerenza, completezza e correttezza dei risultati.

L'obiettivo principale è implementare un processo di verifica per ogni prodotto, assicurando efficienza e accuratezza. Attraverso l'applicazione di tecniche di analisi e test, il processo verifica che i prodotti soddisfino i requisiti specificati. Seguire procedure definite, adottare criteri affidabili e validare il prodotto al termine della verifica sono elementi chiave per garantirne il corretto sviluppo.

La stabilità del prodotto, risultato del processo di verifica, è fondamentale per agevolare il passaggio verso la successiva fase di validazione, assicurando complessivamente la qualità del software.

3.2.2) Descrizione

La verifica rappresenta un processo affidato a un team di verificatori, che si estende a tutti i processi in corso al fine di garantire l'aderenza agli standard stabiliti. Questo procedimento non è un evento isolato, ma piuttosto un ciclo continuo che si ripete periodicamente, adattandosi alle mutevoli esigenze del progetto nel corso del tempo.

Il fulcro di questo processo è il Piano di Qualifica, un documento dettagliato che traccia il percorso della verifica. Esso delinea chiaramente gli obiettivi da raggiungere, i criteri di accettazione da rispettare e i metodi specifici che saranno impiegati per condurre la verifica in modo accurato ed efficiente.

La documentazione derivante dalla verifica non è semplicemente un compito burocratico, ma riveste

un ruolo cruciale nell'assicurare la trasparenza, la ripetibilità e la tracciabilità dell'intero processo. Nei seguenti punti, saranno elencate le possibili attività da adottare.

3.2.3) Analisi statica

L'analisi statica rappresenta una metodologia di verifica che prescinde dall'esecuzione del prodotto, fondata su una revisione critica del codice e della documentazione. L'obiettivo primario di questa analisi è verificare la conformità ai vincoli, l'assenza di difetti e la presenza delle proprietà desiderate. Applicabile a qualsiasi prodotto del progetto, l'analisi statica adotta comunemente due metodi di lettura: il walkthrough e l'inspection.

Il walkthrough, una tecnica collaborativa che coinvolge sia il verificatore che l'autore del prodotto, prevede una lettura a pettine della documentazione e del codice sorgente.

L'inspection, preferibile al walkthrough per la sua velocità ed efficienza, si basa su una lista di controllo e consente di identificare tempestivamente e sistematicamente potenziali problematiche.

L'analisi statica costituisce una fase cruciale nel processo di verifica, ed è vantaggiosa nelle prime fasi del progetto quando i documenti sono ancora relativamente semplici e tutte le attività accuratamente documentate.

3.2.4) Analisi dinamica

L'analisi dinamica costituisce una tecnica di verifica del software che si basa sull'esecuzione del codice al fine di individuare difetti o eventuali problemi di funzionamento ed assicurare la qualità del prodotto finale.

Le tecniche principali utilizzate in questa fase sono i test, rappresentati da esecuzioni del codice con insiemi specifici di dati di input, finalizzati a verificare il comportamento atteso del software.

L'efficacia di un test è legata alla sua natura decidibile e ripetibile. Il termine "decidibile" implica che, dati gli stessi input, il test deve produrre sempre lo stesso risultato, mentre il termine "ripetibile" suggerisce che il test può essere eseguito più volte senza che i risultati siano influenzati da fattori esterni. Fattori determinanti per l'efficacia dei test includono la qualità del codice, l'accurata identificazione dei requisiti e la definizione di un dominio di casi di prova adeguato. Quest'ultimo rappresenta l'insieme completo di tutti i possibili casi di prova che possono essere eseguiti sul software, ed è responsabilità del verificatore definirlo in base ai requisiti del software e alla sua complessità.

L'automazione dei test può essere realizzata attraverso l'utilizzo di strumenti specifici come driver, stub e logger. I driver sono componenti attive che guidano il test, mentre gli stub sono componenti passive che simulano parti del sistema non direttamente coinvolte nel test. I logger, invece, registrano i risultati dei test.

Diversi tipi di test sono disponibili, differenziandosi sulla base della dimensione dell'oggetto del test. I test di unità focalizzano singole unità di codice, come funzioni o metodi, mentre i test di integrazione si concentrano sull'interazione tra diverse unità di codice. I test di sistema, invece, esaminano il comportamento del software come sistema completo, mentre i test di integrazione di sistema considerano l'interazione tra diversi sistemi. I test di accettazione, infine, sono eseguiti dal cliente o utente finale per verificare la conformità del software ai requisiti.

3.3) Validazione

3.3.1) Scopo ed aspettative

La validazione costituisce l'essenziale verifica che il prodotto software sia in linea con i requisiti e le aspettative del cliente, rappresentando una fase cruciale nello sviluppo del software.

Questo processo si concentra attentamente su diversi aspetti:

- Conformità ai requisiti: Il prodotto deve soddisfare integralmente tutti i requisiti specificati dal cliente.

-
- Funzionamento corretto: Il prodotto deve operare correttamente, in conformità con la logica di progettazione.
 - Usabilità: Il prodotto deve essere intuitivo e di facile utilizzo.
 - Efficacia: Il prodotto deve dimostrarsi efficace nel soddisfare le necessità dei clienti.

L'aspettativa finale è di giungere ad un prodotto che risponda pienamente ai requisiti identificati ed operi.

3.3.2) Descrizione

Durante la fase di validazione, concentreremo l'attenzione sull'utilizzo dei test precedentemente eseguiti durante l'attività di verifica, dettagliatamente normati nelle sezioni pertinenti delle Norme di Progetto. Con l'esecuzione del test di accettazione concluderemo la validazione del prodotto.

3.4) Gestione della configurazione

3.5) Gestione della qualità

4) Processi organizzativi