

# Norme di progetto

2023-11-11 — v0.0.2



*overture.unipd@gmail.com*

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

---

## Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
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 .....	1
1.1) Scopo del documento .....	1
1.2) Scopo del progetto .....	1
1.3) Glossario .....	1
1.4) Riferimenti .....	1
1.4.1) Riferimenti normativi .....	1
1.4.2) Riferimenti informativi .....	1
2) Processi primari .....	2
3) Processi di supporto .....	2
3.1) Verifica .....	2
3.1.1) Scopo ed aspettative .....	2
3.1.2) Descrizione .....	2
3.1.3) Analisi statica .....	2
3.1.4) Analisi dinamica .....	2
3.2) Validazione .....	3
3.2.1) Scopo ed aspettative .....	3
3.2.2) Descrizione .....	3
3.3) Gestione della configurazione .....	3
3.4) Gestione della qualità .....	3
4) Processi organizzativi .....	3

## 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](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) Verifica**

#### **3.1.1) 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.

#### **3.1.2) 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.

#### **3.1.3) 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.

#### **3.1.4) 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.

## **3.2) Validazione**

### **3.2.1) 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.

### **3.2.2) 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.

## **3.3) Gestione della configurazione**

## **3.4) Gestione della qualità**

# **4) Processi organizzativi**