



8Lab Solutions - Progetto "Soldino"

Piano di Qualifica

Versione	1.0.0
Approvazione	
Redazione	
Verifica	
Stato	
Uso	Esterno
Destinato a	8Lab Solutions Prof. Tullio Vardanega Prof. Riccardo Cardin

Descrizione

Questo documento descrive le operazioni di verifica e validazione seguite durante lo svolgimento del progetto Soldino.

8labsolutions@gmail.com

Tabelle delle modifiche

Versione	Data	Nominativo	Ruolo	Descrizione
0.0.3				.
0.0.2	2018-01-02	Sara Feltrin		Sistemata struttura del documento e inizia stesura §4.
0.0.1	26-12-2018	Paolo Pozzan	<i>Analista</i>	Creato documento L ^A T _E X e stesura §1.

Indice

Elenco delle figure

Elenco delle tabelle

Introduzione

Scopo del documento

Questo documento ha lo scopo di mostrare le strategie di verifica e validazione adottate al fine di garantire la qualità di prodotto e di processo. Per raggiungere questo obiettivo viene applicato un sistema di verifica continua sui processi in corso e sulle attività svolte. In questo modo è quindi possibile rilevare e correggere all'istante eventuali anomalie, riducendo al minimo lo spreco delle risorse.

Scopo del prodotto

Lo scopo del prodotto è quello di realizzare un software, in particolare un sito internet, che consenta il monitoraggio automatico dell'IIVA, ovvero assiste il governo e gli utenti nell'esecuzione di operazioni come liquidazione, versamento e rimborso, e permetta l'acquisto di oggetti tramite una valuta denominata "Cubit"_G.

Ambiguità

All'interno dei documenti alcuni termini presentano significati ambigui a seconda del contesto, fraintendibili, o che necessitano di una descrizione più approfondita. Per evitare questa ambiguità è stato creato il documento "Glossario" volto a fare chiarezza, ponendo a fianco di ogni termine il suo preciso significato. Questi termini sono pertanto marchiati con una "G" a pedice per ogni loro occorrenza all'interno di tutti i documenti.

Riferimenti

Riferimenti normativi

- **ISO/IEC 12207**
- **Testo del capitolato**
<https://www.math.unipd.it/tullio/IS-1/2018/Progetto/C6.pdf>;

Riferimenti informativi

- Guide?(vedi Pro-tech)
- **ISO/IEC 9126**
https://en.wikipedia.org/wiki/ISO/IEC_9126;
- **Indice di Gulpease**
https://it.wikipedia.org/wiki/Indice_Gulpease

Qualità di processo

Per ricercare qualità nello svolgimento del progetto si adoperano dei processi. Inizialmente, tali processi sono stati scelti tra quelli proposti nello standard ISO/IEC/IEEE 12207:2017; successivamente sono stati semplificati.

Il risultato sono i processi esposti a seguito.

Processi Primari

Processo di Analisi dei Requisiti

Mediante il processo le informazioni ottenute dalle varie fonti sono trasformate in forma di casi d'uso e requisiti. Questa forma fornisce una descrizione dettagliata del sistema e definisce il funzionamento e le caratteristiche di ogni sua parte.

2.1.1.1 Obiettivi

- definire casi d'uso e requisiti;
- ottenere la loro approvazione;
- tracciare il loro cambiamento nel tempo.

2.1.1.2 Strategia Tracciare i requisiti del sistema.

2.1.1.3 Metriche

Numero di requisiti obbligatori soddisfatti Indica la percentuale di requisiti obbligatori soddisfatti.

- Misurazione
- Valore preferibile
- Valore accettabile

Processo di Progettazione Architettuale

Tradurre i requisiti in un modello architettuale del sistema. Il modello è ad alto livello di dettaglio: in esso il sistema è costituito da macro-componenti, utili a capire il funzionamento delle parti, ma non ancora realizzabili nella pratica.

Nel processo vanno considerati in particolare il tipo di software da produrre, le caratteristiche desiderate e i suoi requisiti non funzionali per scegliere l'architettura più adatta.

2.2.0.1 Obiettivi

- valutare il tipo di software da produrre, le caratteristiche desiderate, i casi d'uso e i requisiti da soddisfare;
- valutare i modelli architettureali secondo il punto precedente e scegliere il più adeguato;
- individuare nel modello i macro-componenti del sistema;
- comprendere e definire le relazioni tra i componenti.

2.2.0.2 Strategia Tracciare le componenti del sistema.

2.2.0.3 Metriche

Structural Fan-in SFIN Indica quante componenti utilizzano un dato modulo. Un alto valore indica un alto riuso della componente.

- **Misurazione:** numero intero;
- **Valore preferibile:** ≥ 1 ;
- **Valore accettabile:** ≥ 0 .

Structural Fan-out SFOUT Indica quante componenti vengono utilizzate dalla componente in esame. Un alto valore indica un alto accoppiamento della componente.

- **Misurazione:** numero intero;
- **Valore preferibile:** $= 0$;
- **Valore accettabile:** ≤ 6 .

Processo di Progettazione di Dettaglio

La progettazione di dettaglio segue la progettazione architetturale, e prevede la scomposizione delle macro-componenti in componenti più piccole che sono:

- immediatamente comprensibili;
- strettamente collegate ai requisiti funzionali;
- sviluppabili da un singolo programmatore.

2.3.0.1 Obiettivi Arrivare alla definizione di dettaglio del sistema, cioè scomporlo in unità software dette componenti.

2.3.0.2 Strategia

2.3.0.3 Metriche

Numero di metodi per classe

Numero di parametri per metodo

Processi di Supporto

Processo di Verifica

Il processo consiste nella ricerca e correzione di anomalie nei processi e nei prodotti del progetto, mediante tecniche definite.

2.5.0.1 Obiettivi

- individuare e correggere le anomalie;
- provare che il sistema soddisfi i requisiti.

2.5.0.2 Strategia

- individuare tecniche e strumenti di verifica;
- affinare tecniche e strumenti con l'esperienza;

2.5.0.3 Metriche

Code Coverage Indica il numero di righe di codice percorse dai test durante la loro esecuzione.

- misurazione: Le/Lt ;
- valore preferibile: 100%;
- valore accettabile: 75%.

Processo di Gestione della Documentazione**2.5.1.1 Obiettivi****2.5.1.2 Strategia****2.5.1.3 Metriche**

-
-
-

Processi Organizzativi**Processo di Pianificazione di Progetto****2.7.0.1 Obiettivi****2.7.0.2 Strategia****2.7.0.3 Metriche**

-
-
-

Processo di Gestione della Qualità

2.8.0.1 Obiettivi

2.8.0.2 Strategia

2.8.0.3 Metriche

-
-
-

Processo di Misurazione

2.9.0.1 Obiettivi Scegliere metriche adeguate, effettuare misurazioni basate su tali metriche per migliorare, controllare e garantire la qualità di prodotto e processo.

2.9.0.2 Strategia

- definire gli oggetti di interesse da misurare;
- scegliere metriche appropriate;
- effettuare le misurazioni nel modo più possibile automatizzato;

2.9.0.3 Metriche

-
-
-

Qualità di prodotto

Per valutare la qualità del prodotto il gruppo ha deciso di far riferimento allo standard ISO/IEC 9126_G che definisce le caratteristiche di cui tener conto nel momento in cui si voglia produrre un prodotto di buona qualità. Le caratteristiche sono descritte attraverso dei parametri misurabili che permettono di quantificare il raggiungimento della caratteristica in questione. Di seguito vengono citate le voci che il gruppo ha ritenuto rilevanti in relazione al contesto.

Funzionalità

Capacità del prodotto di fornire funzioni che riescano a soddisfare tutti i requisiti, sia quelli espliciti che quelli impliciti, presenti nell'Analisi dei Requisiti.

Obiettivi

- **Appropriatezza:** il prodotto deve mettere a disposizione un insieme di funzioni conformi agli obiettivi richiesti;
- **Accuratezza:** il prodotto deve fornire risultati attestati con il grado di precisione richiesto;
- **Conformità:** il prodotto deve aderire a determinati standard.

Metriche

3.1.2.1 Completezza dell'implementazione

- La completezza del prodotto e il rispetto dei requisiti viene indicato da una percentuale che si calcola con la seguente formula:

$$C = (1 - \frac{N_{FNI}}{N_{FI}}) \cdot 100$$

dove N_{FNI} indica il numero di funzionalità non implementate e N_{FI} indica il numero di funzionalità individuate dall'analisi.

- Il valore di accettazione di C è 100%, a indicare quindi che tutte le funzionalità, individuate dai requisiti obbligatori, sono state implementate

Affidabilità

Capacità del prodotto di mantenere prestazioni elevate anche in caso di situazioni non normali o critiche.

Obiettivi

- **Maturità:** il prodotto deve evitare che si verifichino errori e malfunzionamenti;
- **Tolleranza agli errori:** il prodotto mantiene alte prestazioni anche in caso di malfunzionamenti o di un uso scorretto.

Metriche

3.2.2.1 Densità errori

- L'abilità del prodotto di resistere a malfunzionamenti viene indicata con una percentuale che si calcola con la seguente formula:

$$M = \frac{N_{ER}}{N_{TE}} \cdot 100$$

dove N_{ER} indica il numero di errori rilevati durante il testing e N_{TE} indica il numero di test eseguiti;

- Il valore preferibile è 0%;
- Un valore accettabile è non superiore al 10%.

Usabilità

Capacità del prodotto di essere di facile comprensione e utilizzo da parte degli utenti.

Obiettivi

- **Comprensibilità:** l'utente deve essere in grado di comprendere le funzionalità offerte dal prodotto e ad utilizzarle;
- **Apprendibilità:** l'utente deve poter imparare facilmente ad utilizzare il prodotto;
- **Attrattività:** il prodotto deve essere piacevole all'utente che ne fa utilizzo;

Metriche

3.3.2.1 Facilità di utilizzo

- La facilità con cui l'utente raggiunge ciò che vuole viene rappresentata tramite il numero di click che sono necessari per arrivare ad esempio alla pagina di fine ordine;
- Il valore preferibile è 10 click;
- Un valore accettabile è non superiore a 15.

3.3.2.2 Facilità di apprendimento

- La facilità con cui l'utente riesce ad imparare ad usare le funzionalità del prodotto viene rappresentata tramite il tempo medio che serve per comprenderle;
- Il valore preferibile è 3 minuti;
- Un valore accettabile è non superiori a 5.

3.3.2.3 Profondità della gerarchia

- La profondità del sito. Un sito per essere facile da utilizzare non deve essere troppo profondo;
- Il valore preferibile è 4;
- Un valore accettabile è inferiore a 7.

Manutenibilità

Capacità del prodotto di essere modificato, includendo correzioni, miglioramenti o adattamenti.

Obiettivi

- **Analizzabilità:** facilità con la quale è possibile analizzare il codice per localizzare un errore;
- **Modificabilità:** capacità del prodotto di permettere l'implementazione di una modifica;

Metriche

3.4.2.1 Facilità di comprensione

- La facilità con cui è possibile comprendere cosa fa il codice può rappresentata dalla seguente formula:

$$R = \frac{N_{L\text{COM}}}{N_{L\text{COD}}}$$

dove $N_{L\text{COM}}$ indica le linee di commento e $N_{L\text{COD}}$ indica le linee di codice;

- Il valore ottimale di R è superiore a 0.20;
- Un valore accettabile è superiore a 0.10.

3.4.2.2 Facilità delle funzioni

- La facilità di un metodo può essere rappresentata dal numero di parametri per metodo: meno parametri ha una funzione più è semplice e intuitiva;
- Il valore preferibile è inferiore o uguale ai 3 parametri per metodo;
- Un valore accettabile è non superiore a 6.

3.4.2.3 Facilità delle classi

- La facilità di una classe può essere rappresentata dal numero di metodi per classe: una classe con pochi metodi ha uno scopo ben preciso e facilmente comprensibile;
- Il valore ottimale di metodi per classe è inferiore a 8;
- Un valore accettabile è non superiore a 15.

Metriche per i documenti

Per assicurare che tutti i documenti prodotti siano leggibili, comprensibili e corretti sotto tutti i punti di vista questi devono rispettare determinate metriche.

Correttezza ortografica

Tutti i documenti, sia quelli in italiano che quelli in inglese, non devono assolutamente contenere errori grammaticali o errori ortografici.

- Saranno accettati solamente documenti privi di errori grammaticali o ortografici.

Indice di Gulpease

Il gruppo si impegna a scrivere documenti facilmente leggibili.

- L'indice di Gulpease fornisce una misura della leggibilità di un documento scritto italiano. La formula che calcola questo indice è la seguente:

$$89 + \frac{330 \cdot (\text{numero delle frasi}) - 10 \cdot (\text{numero delle lettere})}{\text{numero delle parole}}$$

Questa formula produce un risultato compreso tra 0 e 100 dove 100 indica la leggibilità più alta;

- Il valore ottimale sarà superiore a 80;
- Un valore accettabile è superiore a 40.

Specifica dei test

Per produrre software di qualità, il gruppo *8Lab Solutions* definirà dei test per assicurarsi che le unità prodotte funzionino in maniera corretta. Seguirà quindi il tracciamento dei test e il loro esito per mezzo di tabelle che ne semplificheranno la consultazione e che potranno fornire una precisa indicazione degli output prodotti, specificando se il risultato ottenuto sia quello atteso, errato oppure non coerente a quanto fissato in precedenza. Per definire lo stato dei test, vengono utilizzate le seguenti sigle:

- **I**: per indicare che il test è stato implementato;
- **NI**: per indicare che il test non è stato implementato.

Inoltre per lo stato dei test si usano le seguenti abbreviazioni:

- **S**: per indicare che il test ha soddisfatto la richiesta;
- **NS**: per indicare che il test non ha soddisfatto la richiesta.

Tipi di test

Vengono individuate quattro tipologie di test:

- **Test di Unità [TU]**: questi test hanno il compito di verificare le singole unità del software, ovvero le minime componenti del programma che hanno un funzionamento autonomo. Il successo da parte di questi test non implica il corretto funzionamento da parte del software;
- **Test di Integrazione [TI]**: lo scopo di questi test è quello di testare come un gruppo i singoli moduli del software. Essi vengono svolti dopo i Test di Unità e prima dei Test di Sistema;
- **Test di Sistema [TS]**: questi test vengono utilizzati quando il sistema viene installato su una piattaforma e verificano che esso raggiunga gli obiettivi fissati e soddisfi le richieste formulate in partenza;
- **Test di Accettazione [TA]**: essi vengono fatti alla fine per verificare che il prodotto finale soddisfi quanto richiesto dal proponente.

Test di Unità

I test di unità hanno l'obiettivo di verificare il corretto funzionamento della parte più piccola autonoma del lavoro realizzato.

Test di Integrazione

I test di integrazione sono usati per verificare il corretto funzionamento tra le varie unità dell'architettura.

Test di Sistema

I test di sistema sono impiegati per garantire il corretto funzionamento delle componenti dell'intero sistema.

Test di Accettazione

I test di accettazione hanno lo scopo di dimostrare che il software sviluppato soddisfi le richieste del proponente ed essi vengono eseguiti durante il collaudo finale.

Standard di qualità

ISO/IEC/IEE 12207:2017

ISO/IEC 9126