1. Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è quello di fissare tutte le regole e le metodologie da applicare durante la realizzazione del progetto di modo che il gruppo possa lavorare seguendo certi standard ben definiti. In particolare, verranno definiti gli strumenti da utilizzare e le procedure da applicare durante le varie fasi dello sviluppo del software. In tal modo ci sarà una efficiente collaborazione tra i membri.

1.2 Scopo del prodotto

Lo scopo del prodotto è quello di realizzare un’applicazione per dispositivi mobile ed un’interfaccia web in grado di realizzare delle routine personalizzabili dagli utenti tramite l’utilizzo di Alexa *G*. Quest’ultimo è un assistente personale intelligente creato da Amazon ed è in grado di ricevere comandi vocali ed elaborarli.

1.3 Glossario

Nel documento ci sono dei termini con un significato ambiguo a seconda del contesto nel quale sono stati utilizzati. Per ovviare a questo problema è presente il documento Glossario che conterrà una lista di termini con la specifica descrizione del suo significato. Un qualsiasi termine presente nel glossario, verrà indicato in questo documento con una G corsiva in apice che seguirà la parola in questione.

1.3 Riferimenti utili

* Piano di progetto v 1.0.0
* Piano di Qualifica v 1.0.0
* altro che mi viene in mente mentre scrivo...
* Sito del corso di Ingegneria del Software

https://www.math.unipd.it/~tullio/IS-1/2018/Dispense/L08.pdf

2. Processi primari

2.1 Fornitura

Verranno ora trattate le norme che i membri di *duckware* devono rispettare per poter diventare fornitori nei confronti di zero12 e dei committenti, prof Tullio Vardanega e prof Riccardo Cardin.

2.1.1 Studio di fattibilità

In data 16 novembre 2018 è avvenuta presso la Torre Archimede la presentazione dei capitolati d’appalto da parte di alcune aziende interessate nel coinvolgimento degli studenti in un progetto. Il giorno successivo è stata organizzata una riunione del gruppo per discutere di vari argomenti, primo fra tutti la scelta dei capitolati più interessanti da analizzare.

Dopo aver scelto il capitolato per il quale proporsi è stata eseguita un’ulteriore analisi relativa ai rischi ed alle opportunità che è stata riportata all’interno dello Studio di fattibilità 1.0.0. Si tratta di un documento nel quale viene indicato, per ogni capitolato, le motivazioni per le quali il gruppo duckware vorrebbe proporsi come fornitore. In particolare, sono presenti per ogni capitolato:

* Una descrizione del prodotto proposto dall’azienda secondo le regole del capitolato d’appalto *G*
* Una descrizione di quello che sarà l’ambito d’uso del prodotto che ci si propone di creare
* Le tecnologie richieste dal capitolato per la realizzazione del relativo progetto
* Le motivazioni fornite dal *duckware* riguardo l’accettazione o la respinta del capitolato in questione

2.2 Sviluppo

2.2.1 Analisi dei Requisiti

L’analisi dei requisiti ha lo scopo di individuare i requisiti richiesti dal capitolato preso in questione; tali informazioni possono essere reperite da verbali, casi d’uso oppure dai capitolati d’appalto stessi. Il risultato finale di questa analisi porterà alla creazione da parte degli analisti del documento Analisi dei requisiti 1.0.0 che fornirà informazioni riguardo:

* Una stima dei costi
* Descrivere l’obiettivo del lavoro del gruppo e fornire riferimenti precisi ai progettisti
* Fornire una lista dettagliata di funzionalità e requisiti concordati col cliente
* Dare ai verificatori riferimenti e casi d’uso per poter creare ed eseguire dei test mirati ed affidabili
* Rendere il processo di revisione del codice semplice

2.2.1.1 Classificazione di requisiti

Ogni requisito è identificato da un codice che verrà rappresentato secondo quanto segue:

**R.{x}.{y}.{z}**

* **R**: sta per **R**equisito
* **X**: indica l’importanza di tale requisito che può essere 0 (opzionale), 1 (requisito non necessario ma può dare maggiore completezza e definizione) e 2 (obbligatorio in quanto necessario per il funzionamento basilare).
* **Y**: assume valore F se requisito funzionale, Q se di qualità, V se di vincolo e P se presentazionale

2.2.1.1 Classificazione di requisiti

Gli analisti avranno anche cura di identificare i casi d’uso i quali verranno descritti secondo il seguente schema:

**UC[Codice principale].[Codice secondario] – Identificativo**

* **UC ed Identificativo:** Ogni caso d’uso è specificato da una serie di cifre spaziare da un punto. La prima cifra indica il numero del padre, la seconda invece il numero del figlio. Un trattino separerà il tutto dal nome univoco del caso d’uso
* **Attori:** gli attori principali e secondari dell’UC (use case)
* **Scopo e descrizione:** una descrizione riassuntiva del caso d’uso
* **Scenario principale:** usando una lista numerata indica il flusso degli eventi specificando per ognuno a quale caso d’uso si riferisce
* **Pre condizione:** specifica le condizioni vere prima del verificarsi degli eventi del caso d’uso
* **Post condizione:** specifica le condizioni vere dopo del verificarsi degli eventi del caso d’uso

2.2.2 Progettazione

2.2.2.1 Scopo

La progettazione ha il compito di disegnare, attraverso i Progettisti, una soluzione del problema che sia esaustiva per tutti gli stakeholders. Verrà definita la logica del prodotto identificando i componenti e cercando sempre di restare nei costi che sono stati prefissati. In particolare la logica definita dovrà:

* Soddisfare i requisiti *G* dell’Analisi dei requisiti ed adattarsi ai cambiamenti od alle evoluzioni che potenzialmente avverranno nel tempo
* Essere comprensibile e modulare
* Avere la capacità di rispettare le specifiche nel tempo, anche nel caso di malfunzionamenti
* Ridurre al minimo i tempi di manutenzione

2.2.2.2 Diagrammi

Le scelte adottate dai progettisti verranno adeguatamente descritte attraverso l’uso di diagrammi **UML 2.0** per rendere più chiare e meno ambigue le decisioni prese. Verranno in particolare utilizzare le seguenti rappresentazioni:

* **Diagrammi di casi d’uso**: descrivono le funzioni del sistema
* **Diagrammi delle classi**: descrivono gli oggetti e le dipendenze fra essi
* **Diagrammi dei package**: descrivono la dipendenza tra i package (i quali contengono le classi)
* **Diagrammi di sequenza**: descrivono la collaborazione nel tempo degli oggetti
* **Diagrammi di attività**: descrivono la logica procedurale

2.2.2.3 Obiettivi della progettazione

La progettazione garantisce che il prodotto finale soddisfi tutte le richieste emerse e specificate durante l’analisi. Deve garantire la qualità del prodotto sviluppato e cercare di ottimizzare l’uso delle risorse disponibili. Inoltre, una buona progettazione cerca di suddividere nel migliore dei modi i compiti implementativi al fine di ridurre la complessità del problema e, di conseguenza, facilitare il compito dei programmatori.

2.2.3 Codifica

I programmatori dovranno rispettare alcune regole che garantiranno uniformità e coesione al codice che verrà prodotto durante la creazione del progetto. Ci sono sia delle norme globali alle quali i programmatori si devono attenere, indipendentemente dal linguaggio di programmazione scelto, sia delle norme specifiche legate ai linguaggi Delphi, PHP e Javascript.

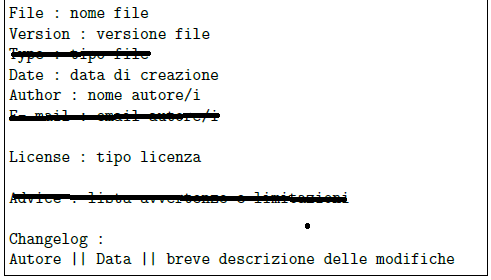
Per chiarezza ogni norma avrà un suo paragrafo composto di titolo, descrizione ed esempio illustrativo di quanto si vuole specificare.

**Convenzioni per i nomi**

* Ogni nome deve essere unico e soprattutto deve essere **appropriato**, ovvero deve descrivere al meglio ciò che rappresenta
* In caso di nome formato dalla composizione di più parole, sarà necessario utilizzare l’underscore come separatore. In alternativa, si potranno separare i termini con una lettera maiuscola. Ad esempio:
  + variable\_name
  + ANOTHER\_EXAMPLE\_FOR\_THIS
  + yetAnotherExample
* Usare le lettere maiuscole quando si intende utilizzare una parola come attributo **const** (costante). Altrimenti, usare lettere minuscole. Per esempio:
  + (costante) THIS\_IS\_AN\_EXAMPLE
  + (non costante) this\_is\_an\_example
  + (non costante) thisIsAnExample

**Convenzioni per la documentazione**

* I commenti ed i nomi presenti nel documento devono tassativamente essere scritti in inglese
* Ogni file dovrà contenere nella sua intestazione, come commento, la seguente struttura:



* La versione va inserita come X.Y dove:
  + X: indica la versione principale ed è da incrementare **solo ed esclusivamente** nel caso in cui si passi ad una versione stabile. Un tale incremento comporta l’azzeramento di Y
  + Y: indica la versione parziale ed un incremento di tale indice rappresenta una verifica o una modifica rilevante che verrà sottoposta alla fase di test.

2.2.3.1 Delphi

In questa sezione vengono indicate delle convenzioni indicate dal team di sviluppo di Delphi 1 (<http://edn.embarcadero.com/article/10280> metti il link sotto) che dovranno essere rispettate nel progretto dagli sviluppatori durante la scrittura del codice.

**Commenti.**

Ogni file sorgente deve iniziare con un blocco di informazioni nell’intestazione come specificato in 2.2.3 – Convenzioni per la documentazione. In particolare, i caratteri dovranno essere contenuti in un riquadro che verrà creato nel seguente modo:

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

{ }

{ File : nome file }

{ Version: versione file }

{ }

{ //... }

{ }

{\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*}

All’interno di metodi, classi, procedure o interfaccie ogni commento su riga singola dovrà essere commentato con il doppio slash, mentre per commenti su più righe si dovrà utilizzare la cosiddetta “struttura a parentesi”. Quest’ultima prevede l’inclusione del testo da commentare fra parentesi graffe separate da uno spazio prima del carattere che le segue o precede. Inoltre il testo dovrà essere allineato.

|  |  |
| --- | --- |
| Commento su riga singola | Commento su più righe |
| //Commento su una riga | |  | | --- | | { Commento su  più righe } | |

**Inclusioni.**

QUESTA PARTE LA FACCIO ENTRO MERCOLEDI CHE TANTO CI METTO POCO, VERRANNO FUORI 10/15 PAGINE

3. Processi di supporto

3.1 Documentazione

3.1.1 Descrizione

Verranno qui discusse le scelte intraprese per la scrittura, verifica ed approvazione della documentazione ufficiale di *duckware*. Queste norme sono obbligatorie per tutti i documenti, i quali verranno elencati nella sotto-sezione “Documenti correnti”.

3.1.2 Ciclo di vita della documentazione

Qualsiasi documento dovrà passare per gli stati di “Sviluppo”, “Verifica” ed “Approvato”. Nel particolare, ciascuno di questi indica una fase precisa:

* **Sviluppo**. Questa fase inizia con la creazione del documento e termina con la conclusione della stesura di tutte le due parti. In questa fase i Redattori aggiungono le parti assegnate usando i ticket.
* **Verifica**. Questa fase inizia dopo l’assegnazione da parte del responsabile. I Verificatori effettueranno i controlli necessari e, in caso di esito positivo, il documento entra automaticamente nella fase “*Approvato*”. In caso di esito negativo sarà necessario che il Responsabile di progetto riassegni il documento ad un Redattore attraverso una nuova fase di Sviluppo.
* **Approvato**: Questa fase coincide con il completamento della parte di Verifica avvenuta con successo nella quale il Verificatore certifica la correttezza. Il documento verrà dunque consegnato al Responsabile il quale lo approverà per il rilascio esterno.

3.1.3 Separazione tra documenti interni ed esterni

Ogni documento dovrà avere una specifica classificazione. Vi saranno sia documenti **interni** in lingua italiana che verranno utilizzati da *duckware* internamente, sia documenti **esterni** che verranno condivisi con la Proponente ed i committenti. Questi ultimi potrebbero essere scritti in lingua Inglese nel caso potesse essere utile al fine di soddisfare le richieste di deploy dell’utente finale.

3.1.4 Nomenclatura dei documenti

Ogni documento, tranne la Lettera di Presentazione ed i Verbali, adotteranno questo schema di nomenclatura:

* **NomeDocumento**: indica il nome del documento e dovrà essere senza spazi con il vincolo di avere una lettera maiuscola all’inizio di ogni parola
* **vX.Y.Z**: indica il numero versionamento.

Nel particolare, il versionamento sarà composto da tre numeri interi, separati da un punto, con il seguente significato:

* **X:** rappresenta il numero di pubblicazioni ufficiali del documento e ad ogni suo incremento corrisponde un azzeramento di Y e Z
* **Y:** rappresenta il numero di verifiche terminate con successo e ad ogni suo incremento corrisponde un azzeramento di Z
* **Z:** rappresenta il numero di modifiche effettuate al documento durante lo sviluppo

Ogni documento sarà redatto all’interno di un file *.tex* e solamente dopo lo stato di “Approvato” verrà generato il relativo PDF che conterrà la versione definitiva approvata dal Responsabile di progetto.

3.1.5 Documenti correnti

Verranno ora esposti i documenti formali, in ordine alfabetico, e divisi per appartenenza (Interno ed Esterno):

**ESTERNO**

* + **[ AR ] - Analisi dei Requisiti:** Questo documento viene redatto agli Analisti dopo che questi ultimi hanno analizzato il capitolato e interagito con il proponente. All’interno dell’analisi dei requisiti ci saranno esposti tutti i requisiti del progetto, inclusi i diagrammi di interazione con l’utente ed i casi d’uso.
  + **[ GL ] – Glossario:** Documento che raccoglie tutti i termini che verranno usati nei documenti formali; serve a disambiguare termini o a facilitarne la comprensione
  + **[ PdP ] – Piano di progetto:** Documento che tratta della pianificazione e dell’analisi della gestione delle risorse di tempo
  + **[ PdQ ] – Piano di qualifica:** Tale documento descrive gli obiettivi che il gruppo sarà tenuto a soddisfare al fine di garantire la qualità del prodotto e del processo

**INTERNO**

* + **[ NdP ] – Norme di progetto:** Documento che descrive gli standard adottati da *duckware* durante lo sviluppo del progetto scelto
  + **[ SdF ] – Studio di fattibilità:** Documento che analizza i pregi ed i punti a sfavore di ogni capitolato con le relative riflessioni che hanno portato *duckware* alla sua scelta finale.

3.1.6 Norme

**3.1.6.1 Struttura dei documenti**

Ogni documento presentato è stato realizzato seguendo uno schema generale che dovrà essere rispettato in ogni documento ufficiale, fatta eccezione della lettera di presentazione e dei verbali. I criteri sono i seguenti:

* **Frontespizio:** sezione presente nella prima pagina di ogni documento e conterrà il logo di *duckware*, il titolo del relativo documento, la versione del documenti, il nome del gruppo ed il nome del progetto.
* **Informazioni sul documento:** si compone di una lista di responsabili, verificatori, redattori del documento e della tipologia d’uso del documento (Interno oppure Esterno [link a sopra]).
* **Diario delle modifiche:** sarà presente nella seconda pagina di ogni documento e conterrà una tabella, ordinata decrescentemente per data, delle modifiche apportate al documento. Nello specifico, ogni riga conterrà: versione, data, descrizione delle modifiche, autore e ruolo
* **Indice delle sezioni:** si tratta di un elenco degli argomenti trattati nel documento che conterrà: titolo, argomento e numero pagina
* **Indice delle tabelle:** sezione che contiene l’elenco delle tabelle e ha la stessa struttura dell’indice delle sezioni
* **Introduzione:** contiene lo scopo del documento, le informazioni sul glossario ed i riferimenti utili normativi ed informativi
* **Contenuto del documento:** tutto ciò che non è stato elencato nei precedenti punti, verrà trattato all’interno del documento

**3.1.6.2 Norme tipografiche**

* **Intestazione:** in ogni pagina del documento, dopo il frontespizio ci sarà il logo di *duckware* sulla sinistra e sulla destra il nome del capitolato in questione
* **Parentesi:** le parentesi tonde descrivono esempi e forniscono sinonimi, le quadre rappresentano uno standard ISO o un riferimento ad un codice definito all’interno del documento stesso.
* **Piè di pagina:** a sinistra c’è il nome del documento con la relativa versione, a destra il numero di pagina
* **Stile del testo**
  + *Corsivo:* dà enfasi ad un ternime o concetto
  + *Grassetto:*per i titoli, sottotitoli o termini all’interno di elenchi o liste
* **Formati:**
  + Date: ogni data verrà formattata secondo il formato dd/mm/yyyy dove dd indica il giorno (a una o due cifre), mm il mese (a una o due cifre) e yyyy l’anno (sempre a quattro cifre).
  + Grassetto: va utilizzato per i titoli die paragrafi e per i titoli di elementi in elendo
  + URI: ogni URI sarà in corsivo e di colore azzurro di modo da essere conformi agli standard degli URI nelle pagine web.
* **Riferimenti informativi:** Ogni riferimento esterno al progetto, come ad esempio guide o software, dovrà essere indicato a piè di pagina
* **Nomi:** sono stati realizzati comandi personalizzati per richiamare la visualizzazione dei seguenti termini:
  + \gruppo visualizza “duckware”
  + (**...** ci va la lista di tutti i comandi che faremo)
* **Componenti grafiche:** sono ammessi formati PNG e JPG ma sono preferibili immagini informato SVG poiche queste ultime preservano una maggiore qualità anche in caso di ridimensionamento.

3.1.7 Norme

È stato creato un template di documentazione che potrà essere impiegato per la realizzazione dei vari documenti ufficiali. Questo template è conforme a tutte le norme di documentazione esposte nelle precedenti sezioni

3.1.8 Glossario

Il glossario è un documento unico che contiene tutte le definizioni, in ordine lessicografico crescente, dei termini presenti nei documenti del progetto che possono avere un significato ambiguo. I termini nel glossario avranno una G corsiva inserita a pedice alla fine della parola in questione. Nel glossario non vi sono duplicati.

Verrà usato il comando {boh?} per contrassegnare una G a pedice mentre per l’inserimento di una parola nel glossario verrà usato \newglossaryentry.

Per facilitare l’inserimento di termini nel glossario, con conseguente aggiornamento in tutti i documenti del progetto con una G a pedice ove necessario, è stato creato un programma GlossaryHelper.exe. Una descrizione di questo tool è presente nella sezione **Strumenti di supporto**.

3.1.8 Ambiente

La scrittura dei documenti dovrà essere realizzata con TexStudio e quindi con la generazione dei file \*.tex. {aggiungere altro se necessario}

3.1.10 Strumenti di supporto

Per facilitare e velocizzare la manutenzione della documentazione è stato realizzato un programma per automatizzare l’esecuzione di certe attività. Sarà possibile utilizzare questo programma in qualsiasi sistema operativo che abbia .NET Framework installato. Il programma GlossaryHelper.exe facilita la manutenzione e l’aggiornamento del glossario e di tutti i suoi documenti; nel particolare, si tratta di un eseguibile dotato di GUI in grado di:

* Caricare il file \*.tex relativo al glossario per inserire e/o rimuovere termini al suo interno. Il programma verificherà la presenza di duplicati nei termini del glossario e, in caso ce ne siano, impedirà l’inserimento.
* In seguito all’aggiunta o alla rimozione di termini dal glossario, sarà possibile aggiornare tutti i file dei documenti. Sarà sufficiente specificare al programma la cartella *root* contenente i documenti da analizzare ed il programma avrà cura di selezionare ogni file \*.tex per procedere all’aggiornamento.

Per poter eseguire il programma è consigliabile avere installata l’ultima versione di .NET Framework; tuttavia il requisito minimo è quello di avere il supporto per .NET Framework 4.7.2 (per C# 7.1).

3.2 Qualità

3.2.1 Descrizione

Questa sezione descrive le procedure per il calcolo dei parametri descritti nel *Piano di qualifica*.

3.2.2 Classificazione dei processi

Per garantire la qualità del lavoro, gli Amministratori hanno suddiviso il lavoro in vari processi che sono stati poi riportati nel *Piano di qualifica*. Dovrà essere rispettata la notazione **PRO[value]**  dove **value** indica il codice univoco del processo tramite un numero intero che parte da 1 ed incrementa per ogni unità.

3.2.3 Classificazione delle metriche

bisogna vederlo assieme dopo aver deciso la struttura degli altri documenti dio bubu

3.3 Configurazione

3.3.1 Classificazione delle metriche

**3.3.1.1** **Descrizione**

Per le parti versionabili del progetto e per i documenti ufficiali si è scelto l’utilizzo di GitLab; la condivisione tramite Google Drive avverrà solo per documenti informali ed altro materiale di supporto durante lo sviluppo del progetto.

**3.3.1.1** **Struttira delle repository**

Nella RR è stata creata una sola repository chiamata “Documentazione” e contenente i documenti suddivisi nel seguente modo:

* Cartella **Esterni**:
  + Analisi dei requisiti
  + Glossario
  + Piano di progetto
  + Piano di qualifica
  + Verbali
* Cartella **Interni**:
  + Glossario
  + Studio di fattibilita
  + Norme di progetto
  + Verbali interni

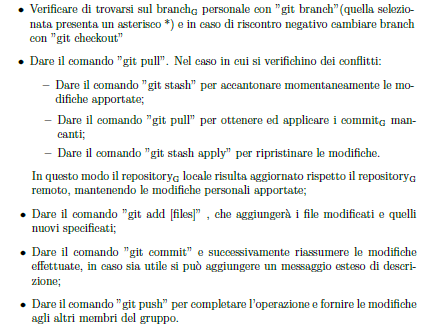
Verranno create successivamente altre repository in base alle necessità che si presenteranno durante lo sviluppo del progetto. All’interno di ogni cartella vi è un file LaTeX che assume il nome del documento ed un altro... {mettere qui cosa}.

**3.3.1.1** **Ciclo di vita dei branch**

Sono stati creati dei branch per poter consentire lo sviluppo in parallelo di uno stesso documento e ciascuno è stato denominato col nome del rispettivo membro. Il merge con il master avviene solamente dopo l’appovazione del documento. I documenti baseline saranno contenuti nel branch master*G*.

**3.3.1.1** **Aggiornamento della repository**

Qualcuno metta qui le procedure, loro avevano messo così però è da rivedere:



3.3.2 Strumenti

* Server git: è stato utilizzato GitLab per l’affidabilità e il supporto a continuous integration.
* Client git: è stata utilizzata l’interfaccia web proposta da GitLab e gli strumenti in linea di comando proposti sempre dallo stesso

3.4 Verifica

3.4.1 Descrizione

In questa sezione verranno descritti gli strumenti ed i metodi impiegati per la verifica del codice e dei documenti durante la loro realizzazione.

3.4.2 Analisi statica

L’analisi statica è una tecnica applicabile al codice ed alla documentazione che permette la verifica di un prodotto individuandone gli errori. Può essere svolta secondo:

* **Inspection:** si applica quando si ha un’idea delle possibili problematiche che si stanno cercando e si effettua facendo una comparazione tra il prodotto e una tabella di possibili errori creata in precedenza
* **Walkthrough:** si applica quando non si sa quali sono le tipologie di errori che si stanno cercando. Occorre ispezionare tutto il codice o il documento per trovare qualsiasi tipo di anomalia

3.4.2 Analisi dinamica

L’analisi dinamica consiste nella creazione ed esecuzione di una serie di test direttamente sul codice, utilizzando anche dei tool già realizzati appositamente per teli scopi. Non è possibile fare analisi dinamica sui documenti.

3.4.4 Verifica diagrammi UML

I verificatori avranno il compito di controllare tutti i diagrammi UML ed assicurarsi che rispettino lo standard UML.

3.4.5 Strumenti

* **Software:** verrà utilizzato DUnitX per la verifica del codice Delphi e PHPUnit per PHP
* **Documenti:** per il controllo dei documenti verranno utilizzate le funzioni integrate a Texstudio assieme a {qualcosa, tipo Travis?} per controllare che non ci siano errori nei file LaTeX. Verrà utilizzato GlossaryHelper per la gestione del glossario e per assicurarsi che non ci siano duplicati.
* **Gestione di processi e feedback:** verrà utilizzato il sistema di issues di GitLab
* **Gestione di tasklist:** potremo creare un semplice sito di 1 o 2 pagine? Molto veloce
* **Analisi dinamica locale:** ??
* **Analisi dinamica remota:** ?? magari Sonarqube

3.5 Validazione

3.5.1 Descrizione

Il processo di validazione ha come scopo quello della verifica del prodotto al fine di assicurarsi che sia conforme a quanto stabilito.

3.5.2 Procedure

L’attività di validazione verrà svolta secondo i seguenti passi:

1. **Tracciamento:** tracciamento dei test con report da parte dei verificatori
2. **Analisi dei risultati:** in seguito al punto 1, spetterà al Responsabile di progetto l’analisi dei report e la decisione di:

* Accettare la validazione e consegnare al proponente i risultati, informandolo sulle modalità di esecuzione della validazione
* Ripetere i test in toto o in parte aggiungendo indicazioni mirate al miglioramento della fase di test che deve essere nuovamente eseguita

4. Processi organizzativi

4.1 Processi di coordinamento

4.1.1 Comunicazione

Verranno ora illustrate le norme che regolano la comunicazione sia tra i membri di *duckware* che con entità esterne quali i proponenti ed i committenti.

**4.1.1.1 Comunicazioni interne**

Le comunicazioni interne di *duckware* avverranno attraverso Slack, il quale consente la creazione di appositi canali tematici in ciascuno dei quali verrà trattato il relativo argomento. In particolare, i canali saranno così suddivisi:

* **Generale.** Verrà discusso di argomenti generali riguardanti l’organizzazione del progetto, la scelta di strumenti di lavoro o per una rapida comunicazione. Verranno anche stabiliti gli argomenti di discussione durante gli incontri.
* **Todo incontri.** In questo canale verranno scritte dal Responsabile tutti gli argomenti da toccare durante la successiva sessione d’incontro nel programma. Inoltre, verrà stilato un breve riassunto di quanto deciso in merito ai punti da discutere.
* **Gitlab monitor.** Grazie all’utilizzo di un bot, ogni modifica fatta da un utente all’interno della repository verrà segnalato all’interno di questo canale
* **Link utili.** In questo canale non sarà possibile scrivere messaggi ma si potranno solamente visualizzare dei collegamenti a risorse enterne che potranno essere utili durante lo sviluppo del progetto.
* **Glossario.** Ogni volta vi sia bisogno di un aggiornamento del glossario verranno qui notificati i termini con le relative definizioni da inserire. Una volta aggiornato il glossario, grazie a GlossaryHelper, i messaggi verranno rimossi per evitare confusione.

Vi sono dei canali, ciascuno avente il nome del documento al quale si riferiscono, che contengono un unico messaggio contente il link al file \*.tex di documentazione. È stato creato inoltre un sottodominio all’interno del server di uno dei membri di *duckware* per fare il backup di documentazione e file.

**4.1.1.1 Comunicazioni esterne**

In questa sottosezione vi sono le norme che regolano le comunicazioni con soggetti esterni a *duckware*, come:

* La proponente **zero12** rappresentata da Dindo Qualcosa, con i quali si intende stabilire un rapporto di collaborazione per la definizione dei requisiti necessari al fine di realizzare il prodotto
* Prof Tullio Vardanega e Prof Riccardo Cardin ai quali verrà fornita la documentazione richiesta in ciascuna revisione di progetto. Con essi si intenderà dialogare per portare un continuo miglioramento al progetto.

Tutte le comunicazioni esterne vengono effettuate attraverso

[suckware.swe@gmail.com](mailto:suckware.swe@gmail.com)

e ogni membro del gruppo ha accesso a tale indirizzo email. I proponenti verranno raggiunti attraverso gli indirizzi da loro forniti:

s.dindo@zero12.it

{Se faremo come 353 un canale slack o telegram con zero12 per comunicare più velocemente, scriverlo qua}. Ci rivolgeremo ai commettenti dando loro del Lei o del Voi.

4.1.2 Riunioni

In questa sotto-sezione definiremo le norme relative alle riunioni interne ed esterne. Durante lo svolgimento di ogni riunione ci sarà un responsabile dell’incontro che si prenderà cura di far rispettare tutti i punti di discussione su cui riflettere. Il rapporto finale di quanto discusso verrà poi inserito all’interno del Verbale di Riunione.

**4.1.2.1 Verbale di riunione**

Il responsabile d riunione redigerà il Verbale di Riunione secondo il seguente schema:

* **Frontespizio.** Il frontespizio sarà formato dall’indicazione della tipologia del verbale, ovvero Interno o Esterno, e dalla data nella quale tale incontro è avvenuto.
* **Registro modifiche.** Nella seconda pagina sarà presente un registro delle modifiche relativo a tutti i cambiamenti apportati al verbale prima di giungere alla sua versione finale.
* **Informazione sulla riunione.** In questa sezione ci saranno:
  + Lo scopo della riunione: il motivo per il quale si è deciso di fare una riunione;
  + Data e luogo: quando e dove la riunione si è tenuta;
  + Durata: include l’orario di inizio e l’orario di fine della riunione, in formato ventiquattro ore, ad esempio 17.40;
  + Partecipanti alla riunione: nel caso di riunione esterna, verranno prima elencati i committenti/proponenti, altrimenti verrà stilata una lista di partecipanti
* **Ordine del giorno:** elenco puntato degli argomenti da discutere
* **Resoconto:** riassunto redatto dal responsabile dell’incontro secondo quanto stabilito dai punti dell’ordine del giorno. Dovrà essere inserito in grassetto il titolo di ogni punto trattato e poi su una nuova riga la descrizione di quanto è stato deciso.

Ogni file di un verbale verrà chiamato nel seguente modo: **verbale\_DATA** dove **DATA** è la data nel quale è stato svolto l’incontro. Ad esempio un titolo valido potrebbe essere *verbale\_12-12-2018*.

**4.1.2.2 Riunioni interne**

La partecipazione alle riunioni interne è rivolta solo ed esclusivamente ai membri di *duckware* ed il Responsabile di progetto ha il compito di stilare l’ordine del giorno. Gli incontri avverranno ogni giovedì a partire dalle ore 13.00. I partecipanti dovranno essere puntuali alle riunioni e comunicare con quanto più anticipo, se possibile, eventuali ritardi o problemi di presenza. Una riunione può avvenire solo se ci sono almeno 5 partecipanti su 7.

**4.1.2.2 Riunioni esterne**

Le riunioni esterne coinvolgono sia la Proponente che i membri di *duckware*; all’interno del verbale esterno relativo all’incontro verranno inseriti tutti gli argomenti trattati. Le riunioni verranno effettuate sede della proponente oppure tramite chiamata di gruppo su programmi di video conferenza come Skype. Quando possibile, gli incontri avverranno presso Torre Tullio Levi cita.

4.2 Processi di pianificazione

**4.2.1 Ruoli di progetto**

La realizzazione del progetto è il prodotto finale di una collaborazione coesa ed organizzata da parte dei membri del gruppo. Ogni membro avrà il suo ruolo all’interno di *duckware* e ciò rifletterà le rispettive figure che ci sono all’interno di una azienda. A rotazione ogni membro del gruppo ricoprirà questi ruoli

* Responsabile di progetto
* Amministratore
* ANALista
* Progettista
* Programmatore
* Verificatore

Il cambio del ruolo avverrà in modo tale da poter garantire un costante livello di qualità. Il responsabile avrà l’onere di assegnare i ruoli in modo ragionato al fine di non creare situazioni ambigue. Ad esempio, non sarà possibile che un verificatore debba verificare dei documenti che sono stati creati dallo stesso in quanto non ci sarebbe una sufficiente analisi critica.

**4.2.1.1 Responsabile di progetto**

Il Responsabile di progetto o Project manager partecipa al progetto per tutta la sua durata e su di esso ricadono tutte le responsabilità di scelta ed approvazione. Inoltre è colui che rappresenta *duckware* nei confronti della Proponente e dei committente. Egli dovrà:

* Approvare i documenti
* Elaborare piani, scadenze e coordinate le attività del gruppo
* Relazionarsi con il controllo di qualità del progetto
* Approvare l’offerta

**4.2.1.2 Amministratore**

L’amministratore la figura fondamentare del gruppo poiché deve garantire l’efficienza e l’attività del gruppo. Esso dovrà occuparsi dell’organizzazione dell’ambiente di lavoro redigendo i documenti che regolato le attività d verifica e di lavoro. Infatti le sue mansioni sono:

* Redigere le norme di progetto e aiutare nella scrittura del Piano di Progetto
* Assicurarsi che la documentazione sia corretta ed approvata
* È responsabile della redazione di piani e procedure relativi alla Gestione per la Qualità

**4.2.1.3 Analista**

L’attività degli analisti è necessaria affinchè il progetto possa essere realizzato; essi dovranno analizzare il problema e comprenderlo a pieno al fine di ridurre la probabilità di realizzare gravi errori di progettazione. Dovranno occuparsi nello specifico di:

* Realizzare lo studio di fattibilità, l’analisi dei requisiti e verificare le implicazioni di costi e qualità
* Studiare e definire al meglio, senza ambiguità, il problema da risolvere per capire cosa deve essere realizzato in base ai requisiti richiesti
* Modellare il problema ed effettuare la ripartizione dei requisiti

**4.2.1.4 Progettista**

Il progettista è il responsabile delle attività di progettazione, le quali consistono nella definizione di una soluzione accettabile per ogni stakeholder coinvolto. I suoi obiettivi sono:

* Definizione della struttura logica del prodotto in modo che sia ottimizzata e quindi facile da mantenere
* Suddivisione del sistema in problemi di complessità ridotta e trattabili al fine di rendere il lavoro di codifica più facile da realizzare per i programmatori e più facile da verificare per i verificatori

**4.2.1.5 Programmatore**

Il programmatore è responsabile del processo di codifica che porta alla realizzazione del prodotto. Il suo compito è quello di implementare attraverso specifici linguaggi quanto è stato definito dal Progettista nell’architettura del progetto. Esso dovrà:

* Scrivere codice propriamente documentato, indentato e versionato
* Scrivere il manuale utente in modo che contenga il minor nuero possibile di ambiguità
* Creare le componenti per la verifica e la validazione del codice

**4.2.1.6 Verificatore**

Il verificatore deve essere presente durante tutta la durata del progetto in quanto si occuperà delle attività di verifica. Esse sono:

* Redazione del piano di qualifica. Al suo interno saranno presenti tutte le prove effettuate ed i risultati ottenuti
* Accertamento che le attività di processo non abbiano generato degli errori

**4.2.1.7 Cambio di ruoli**

Ciascun membro di *duckware* ricoprirà ognuno dei ruoli sopra riportati e la rotazione avverrà secondo le seguenti regole:

* Sarà obbligatorio tenere in considerazione gli interessi e gli impegni del singolo al fine di poter garantire un’esecuzione ottima
* Ogni membrò non potrà mai effettuare la verifica di un’opera svolta dallo stesso poiché potrebbero sorgere dei conflitti di interesse
* Il trasferimento di ruolo dovrà avvenire in modo ottimale e per tali motivi sarà necessario redigere un breve documento informale all’interno di *duckware* nel quale ci saranno commenti da parte del precedente assegnatario di quel ruolo.

4.2 Strumenti

**4.4.1 Pianificazione**

Per gestire al meglio la pianificazione del progetto, *duckware* ha deciso di usare Asana (intanto lo metto, poi decidiamo). Si tratta di un servizio cloud che crea dei task ed assegna ad ogni membro un periodo temporale entro il quale svolgere il proprio operato. In questo modo sarà possibile controllare lo stato di avanzamento dei vari task se essi siano stati completati o se ancora in fase di elaborazione.

**4.2.2 Comunicazione**

La comunicazione tra i membri di *duckware* avviene tramite l’utilizzo di Slack. Questa è un’applicazione di messaggistica apposita per i gruppo di lavoro che supporta anche l’utilizzo di bot che possono interagire con sistemi di controllo di versione come GitHub.

**4.2.2 Diagrammi di Gantt**

I diagrammi di Gantt sono stati creati con GanttProject in quanto quest’ultimo è gratuito e multi piattaforma.

**4.4.4 Clacolo del consuntivo**

Gli strumenti utilizzati dal Reponsabile di progetto sono stati Microsoft Office Excel ed Instagantt per Asana

4.3 Formaziome

**4.3.1 Formazione dei membri del gruppo**

Tutti i membdi di *duckware* devono studiare autonomamente le tecnologie che verranno utilizzare durante il processo di creazione del progetto. Verranno realizzare delle guide informali per uso interno, preferibilmente dai membri con maggiore esperienza nel relativo campo, per facilitare la formazione dei membri di *duckware* che presentano lacune.

**4.3.2 Guide e materiale usato**

La documentazione di riferimento comprende quanto indicato nella seguente lista più tutto ciò che è già stato menzionato all’interno della sottosezione *Riferimenti Informativi*:

* GitLab: https://gitlab.com/help
* LaTeX: https://www.latex-project.org;