

Gestione quaderno dei compiti e valutazioni

Titolo del progetto: Gestione quaderno dei compiti e valutazioni
Alunno/a: Lucas Previtali
Classe: I4AA
Anno scolastico: 2019/2020
Docente responsabile: Ugo Bernasconi

| | | |
|-------|---|----|
| 1 | Introduzione..... | 3 |
| 1.1 | Informazioni sul progetto..... | 3 |
| 1.2 | Abstract..... | 3 |
| 1.3 | Scopo..... | 3 |
| | Analisi..... | 4 |
| 1.4 | Analisi del dominio..... | 4 |
| 1.5 | Analisi e specifica dei requisiti..... | 4 |
| 1.6 | Use case..... | 6 |
| 1.7 | Pianificazione..... | 6 |
| 1.8 | Analisi dei mezzi..... | 6 |
| 1.8.1 | Software..... | 6 |
| 1.8.2 | Hardware..... | 6 |
| 2 | Progettazione..... | 7 |
| 2.1 | Design dell'architettura del sistema..... | 7 |
| 2.2 | Design dei dati e database..... | 7 |
| 2.3 | Design delle interfacce..... | 7 |
| 2.4 | Design procedurale..... | 7 |
| 3 | Implementazione..... | 8 |
| 4 | Test..... | 8 |
| 4.1 | Protocollo di test..... | 8 |
| 4.2 | Risultati test..... | 9 |
| 4.3 | Mancanze/limitazioni conosciute..... | 9 |
| 5 | Consuntivo..... | 9 |
| 6 | Conclusioni..... | 9 |
| 6.1 | Sviluppi futuri..... | 9 |
| 6.2 | Considerazioni personali..... | 9 |
| 7 | Bibliografia..... | 9 |
| 7.1 | Bibliografia per articoli di riviste..... | 9 |
| 7.2 | Bibliografia per libri..... | 9 |
| 7.3 | Sitografia..... | 9 |
| 8 | Allegati..... | 10 |

1 Introduzione

1.1 Informazioni sul progetto

Progetto svolto da: Lucas Previtali

Mandante del progetto: Ugo Bernasconi

Docente Responsabile: Ugo Bernasconi

Scuola: Arti e Mestieri Trevano

Sezione: Informatica

Classe: I4AA

Data d'inizio: 03.09.2019

Termine della consegna: 20.12.2019

1.2 Abstract

E' una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all'utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

- **Background/Situazione iniziale**
- **Descrizione del problema e motivazione:** Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
- **Approccio/Metodi:** Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche...)? Quale è stata l'entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
- **Risultati:** Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

As the size and complexity of today's most modern computer chips increase, new techniques must be developed to effectively design and create Very Large Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.

1.3 Scopo

Lo scopo del progetto (scopi didattici/scopi operativi). Dovrebbe descrivere il mandato, ma non vanno ricopiate le informazioni del quaderno dei compiti (che va invece allegato).

Analisi

1.4 Analisi del dominio

Questo capitolo dovrebbe descrivere il contesto in cui il prodotto verrà utilizzato, da questa analisi dovrebbero scaturire le risposte a quesiti quali ad esempio:

- Background/Situazione iniziale
- Quale è e come è organizzato il contesto in cui il prodotto dovrà funzionare?
- Come viene risolto attualmente il problema? Esiste già un prodotto simile?
- Chi sono gli utenti? Che bisogni hanno? Come e dove lavorano?
- Che competenze/conoscenze/cultura posseggono gli utenti in relazione con il problema?
- Esistono convenzioni/standard applicati nel dominio?
- Che conoscenze teoriche bisogna avere/acquisire per poter operare efficacemente nel dominio?
- ...

1.5 Analisi e specifica dei requisiti

Il progettista, dopo aver ricevuto il mandato, in collaborazione con il committente redige una lista di requisiti. Durante questi incontri, tramite interviste (da inserire nei diari), il progettista deve cercare di rispondere alle seguenti domande:

- Quali sono i bisogni del committente?
- Quali funzioni deve svolgere il prodotto?
- Come devono essere implementate?
- L'utente, come vorrebbe/dovrebbe interagire con il prodotto?
- Come verrà utilizzato il prodotto?
- Che tipo di interfaccia si immagina?
- Che prestazioni minime deve fornire il prodotto?
- Che grado di sicurezza deve avere il prodotto?
- ...

In base alla lista dei requisiti e all'analisi degli stessi, il progettista redige una *specifica dei requisiti* in cui elenca e descrive in modo dettagliato quali sono le funzionalità che il prodotto fornirà. La specifica dovrebbe essere abbastanza dettagliata da poter essere utilizzata come base per lo sviluppo, ma non troppo; ad esempio non dovrebbe contenere dettagli di implementazione, o definizioni dettagliate dell'interfaccia grafica a meno che questi non siano considerati cruciali. Non si deve scordare che i requisiti non rappresentano delle attività bensì delle caratteristiche che il prodotto dovrà possedere.

| ID: REQ-012 | |
|-----------------|--|
| Nome | Creazione interfaccia banca dati |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Si necessitano i permessi di root / Dipende dal requisito REQ-001 (Creazione DB) |
| Sotto requisiti | |
| 001 | Si necessita una maschera di login |
| 002 | Si dovranno poter immettere nuovi allievi |
| 003 | Dovrà essere possibile la ricerca di allievi |

Spiegazione elementi tabella dei requisiti:

ID: identificativo univoco del requisito

Nome: breve descrizione del requisito

Priorità: indica l'importanza di un requisito nell'insieme del progetto, definita assieme al committente. Ad esempio poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

Versione: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l'ultima versione, mentre le vecchie dovranno essere inserite nei diari.

Note: eventuali osservazioni importanti o riferimenti ad altri requisiti.

Sotto requisiti: elementi che compongono il requisito.

1.6 Use case

I casi d'uso rappresentano l'interazione tra i vari attori e le funzionalità del prodotto.

1.7 Pianificazione

Prima di stabilire una pianificazione bisogna avere almeno una vaga idea del modello di sviluppo che si intende adottare. In questa sezione bisognerà inserire il modello concettuale di sviluppo che si seguirà durante il progetto. Gli elementi di riferimento per una buona pianificazione derivano da una scomposizione top-down della problematica del progetto.

La pianificazione può essere rappresentata mediante un diagramma di Gantt:

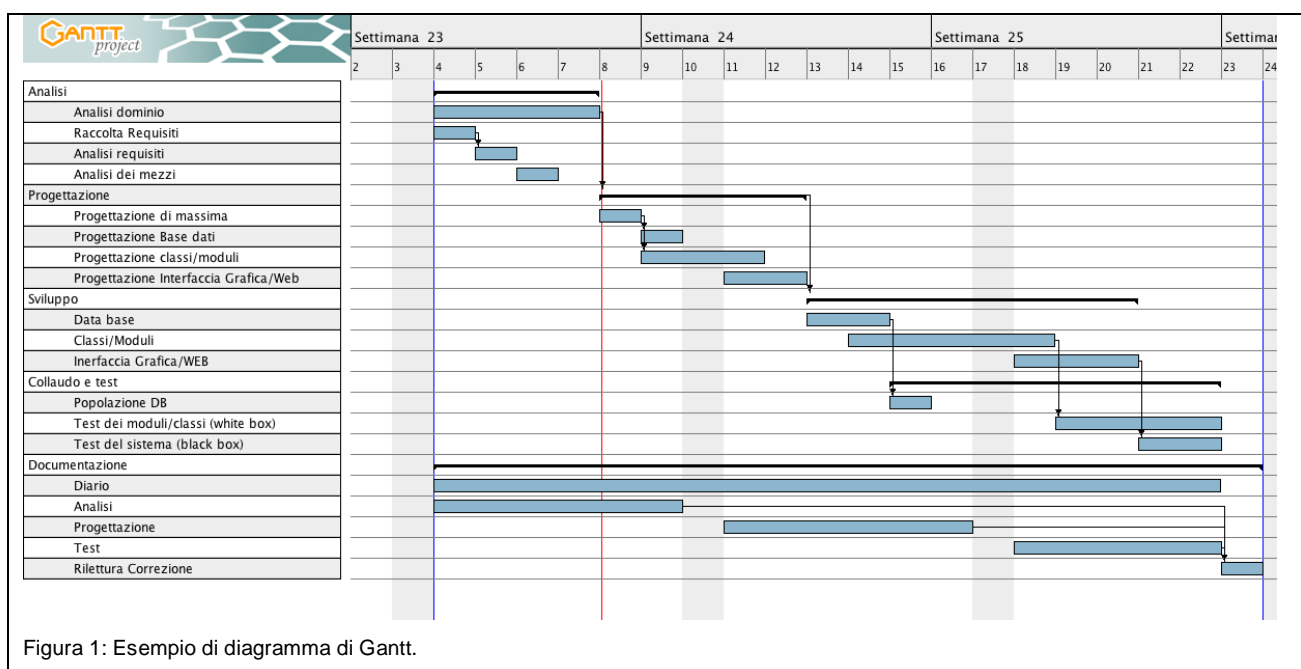


Figura 1: Esempio di diagramma di Gantt.

Se si usano altri metodi di pianificazione (es scrum), dovranno apparire in questo capitolo.

1.8 Analisi dei mezzi

Elencare e descrivere i mezzi disponibili per la realizzazione del progetto. Ricordarsi di sempre descrivere nel dettaglio le versioni e il modello di riferimento.

1.8.1 Software

SDK, librerie, tools utilizzati per la realizzazione del progetto e eventuali dipendenze.

1.8.2 Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che hw sarà disponibile durante lo sviluppo?

2 Progettazione

Questo capitolo descrive esaurientemente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all'esecutore di evitare fraintendimenti e imprecisioni nell'implementazione del prodotto.

2.1 Design dell'architettura del sistema

Descrive:

- La struttura del programma/sistema lo schema di rete...
- Gli oggetti/moduli/componenti che lo compongono.
- I flussi di informazione in ingresso ed in uscita e le relative elaborazioni. Può utilizzare *diagrammi di flusso dei dati* (DFD).
- Eventuale sitemap

2.2 Design dei dati e database

Descrizione delle strutture di dati utilizzate dal programma in base agli attributi e le relazioni degli oggetti in uso.

Schema E-R, schema logico e descrizione.

Se il diagramma E-R viene modificato, sulla doc dovrà apparire l'ultima versione, mentre le vecchie saranno sui diari.

2.3 Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell'interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

2.4 Design procedurale

Descrive i concetti dettagliati dell'architettura/sviluppo utilizzando ad esempio:

- Diagrammi di flusso e Nassi.
- Tabelle.
- Classi e metodi.
- Tabelle di routing
- Diritti di accesso a condivisioni ...

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

3 Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato. Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l'inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d'uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

4 Test

4.1 Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l'adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

| | | | |
|--------------------------|--|--------------|--|
| Test Case: | TC-001 | Nome: | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| Riferimento: | REQ-012 | | |
| Descrizione: | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| Prerequisiti: | Store on local PC: Profile_1.2.001.xml (appendix h-n) and Cards_1.2.001.txt (appendix h-n). PIN (OTA_VIEW_PIN_PUK_KEY) and ADM (OTA_VIEW_ADM_KEY) user right not set. | | |
| Procedura: | <ol style="list-style-type: none"> Go to "Cards manager" menu, in main page click "Import Profiles" link, Select the "1.2.001.xml" file, Import the Profile Go to "Cards manager" menu, in main page click "Import Cards" link, Select the "1.2.001.txt" file, Delete the cards, Select the "1.2.001.txt" file, Import the cards Research the "41795924770" Card, Click the imsi card link Check the card details Execute the SQL: <pre>SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kickey), rawtohex(chv), rawtohex(dap) FROM otacardkey a where imsi='340041795924770' ORDER BY keyset;</pre> | | |
| Risultati attesi: | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

4.2 Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l'errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l'errore con eventuali ipotesi di correzione.

4.3 Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

5 Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gantt consuntivo).

6 Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un'aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

6.1 Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

6.2 Considerazioni personali

Cosa ho imparato in questo progetto? ecc

7 Bibliografia

7.1 Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell'autore o degli autori, o nome dell'organizzazione,
2. Titolo dell'articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell'articolo,

7.2 Bibliografia per libri

1. Cognome e nome (o iniziali) dell'autore o degli autori, o nome dell'organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell'editore,
5. Anno di pubblicazione,
6. ISBN.

7.3 Sitografia

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

Esempio:

- <http://standards.ieee.org/guides/style/section7.html>, *IEEE Standards Style Manual*, 07-06-2008.

8 Allegati

Elenco degli allegati, esempio:

- Diari di lavoro
- Codici sorgente/documentazione macchine virtuali
- Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
- Documentazione di prodotti di terzi
- Eventuali guide utente / Manuali di utilizzo
- Mandato e/o Qdc
- Prodotto
- ...