

Project Plan - Santorini

1. Introduzione

Nell'introduzione al piano di progetto, vengono forniti il background e la storia del progetto, insieme ai suoi obiettivi, i risultati del progetto, i nomi delle persone responsabili e una sintesi del progetto.

L'obiettivo è quello di creare una versione digitale con GUI del gioco da tavolo Santorini, utilizzando i metodi di sviluppo e le regole dettate dall'ingegneria del software in maniera professionale e seguendo ciò che si è studiato nel corso.

Le persone responsabili del progetto sono:

- Andrei Petrisor (1085993)
- Andrea Rusconi (1086646)
- Uriel Fumagalli (1085169)
- Antonio Radu (1085992)

TODO: da rifinire una volta finito il progetto

2. Modello di processo

Nel Capitolo 1 è stato introdotto un semplice modello del ciclo di vita per discutere le varie attività da affrontare in un progetto di sviluppo software. Esistono molte varianti di questo modello di processo, alcune delle quali sono discusse nel Capitolo 3. Per ogni progetto si deve decidere l'esatto modello di processo da seguire: quali attività intraprendere, quali pietre miliari possono essere identificate, come si accerta se tali pietre miliari vengono raggiunte e quali sono i percorsi critici. Diversi tipi di progetti hanno caratteristiche diverse e quindi richiedono modelli di processo diversi.

Utilizzeremo il framework di sviluppo AGILE SCRUM. Organizzeremo lo SCRUM seguendo degli sprint di durata circa una settimana. Ogni giorno avremo un “daily scrum” utilizzando WhatsApp o in chiamata su Discord oppure avendo uno “stand up meeting” di persona quando possibile.

Pietre miliari:

- a. Creazione vari UML
- b. Implementazione classi e metodi con papyrus (Versione **alpha**)
- c. Creazione parte server con GUI (Vaadin)
 - i. Gestione connessioni client/server
- d. Implementazione regole gioco base (da papyrus) sul server (Versione **beta**)
- e. Aggiunta “god powers” semplici (Versione **1.0**)
- f. Implementazione leaderboard
- g. (Opzionale) Aggiunta “god powers” avanzati

3. Organizzazione del progetto

Il rapporto del progetto con altri enti e l'organizzazione del progetto stesso sono trattati in questa rubrica. Il progetto avrà una relazione con l'organizzazione utente, l'organizzazione madre ed eventualmente con altre organizzazioni. I

potenziali utenti saranno, di volta in volta, coinvolti nel progetto. Il piano del progetto deve indicare quali informazioni, servizi, risorse e strutture devono essere fornite dagli utenti e quando devono essere forniti.

All'interno del team di progetto possono essere identificati vari ruoli: project manager, tester, programmatore, analista, ecc. È necessario delineare chiaramente questi ruoli e identificare le responsabilità di ciascuno di essi. Se ci sono lacune nelle conoscenze richieste per ricoprire uno di questi ruoli, è necessario identificare la formazione e l'istruzione necessarie per colmare queste lacune. Diverse forme di organizzazione del team sono discusse nel Capitolo 5 .

Il cliente richiede la versione digitale del gioco in scatola “Santorini”, accessibile in rete locale, per poter giocare con i propri compagni di stanza. Richiede che il programma possieda le regole complete del gioco in scatola e le prime due di queste difficoltà:

1. Gioco senza “god powers” (Difficoltà “Easy”)
2. Gioco con 10 “god powers” semplici (Difficoltà “Normal”)
3. Gioco con 20 “god powers” avanzati (Difficoltà “Advanced”). - opzionale

Inoltre il cliente vuole poter vedere una classifica nella pagina principale (Home) del gioco. - opzionale

Product owner: Uriel Fumagalli

SCRUM master: Andrei Petrisor

Developer: Uriel Fumagalli, Andrei Petrisor, Andrea Rusconi, Antonio Radu

4. Standard, linee guida, procedure

I progetti software sono grandi progetti. Di solito, molte persone sono coinvolte. Occorre quindi una forte disciplina di lavoro, in cui ogni persona coinvolta segua gli standard, le linee guida e le procedure concordate. Oltre ad essere dichiarati sulla carta, molti di questi possono essere supportati o applicati da strumenti. Di estrema importanza sono gli accordi chiari sulla documentazione: quando consegnare la documentazione, come valutare la qualità della documentazione, come garantire l'aggiornamento della documentazione? In larga misura, questi standard e procedure saranno descritti in documenti separati, come il Piano di controllo della configurazione o il Piano di garanzia della qualità.

Per esempio un progetto potrebbe dichiarare di seguire le convenzioni per la programmazione Java come definite da Oracle

Data rilascio gioco: ...

Linguaggio di programmazione: Java

Creazione GUI WebApp: Vaadin

Standard di programmazione: Oracle

Software creazione classi UML: Papyrus

IDE: Eclipse

TODO: rifinire quando iniziamo a scrivere il programma

5. Attività di gestione

sono guidate da obiettivi e priorità fissati per il progetto. Ad esempio, la direzione dovrà presentare relazioni periodiche sullo stato e sullo stato di avanzamento del progetto. Dovrà anche seguire alcune priorità nel bilanciamento di requisiti, tempi e costi

Alla fine di ogni sprint avremo una riunione dove aggiorneremo lo stato dello sviluppo e analizzeremo il lavoro compiuto nello sprint (Sprint Review), insieme a una discussione sui possibili miglioramenti applicabili allo sviluppo del progetto (Sprint Retrospective).

Requisiti:

- Obbligatori
 - Gioco funzionante
 - Regole complete difficoltà “Normal”
 - 10 god powers semplici
- Opzionali
 - Aggiunta god powers avanzati
 - Aggiunta leaderboard

Tempi: è richiesta la consegna del programma funzionante entro 5 giorni dalla prova d'esame

Costi: I nostri costi non saranno monetari ma misurati in ore di lavoro per sviluppatore.

6. Rischi

I potenziali rischi devono essere identificati il prima possibile. Ci saranno sempre dei rischi: l'hardware potrebbe non essere consegnato in tempo, il personale qualificato potrebbe non essere disponibile quando necessario, le informazioni critiche potrebbero mancare quando è necessario e così via. È piuttosto ingenuo supporre che un progetto di sviluppo software funzioni sempre senza intoppi. Anche in campi consolidati, come quello delle costruzioni, la vita è bella e andrà sempre tutto bene 😊. Si dovrebbe diagnosticare precocemente

i rischi di un progetto software e fornire misure per affrontarli; vedere anche il capitolo 8 .

Più incerti sono i vari aspetti del progetto, maggiori sono i rischi.

I potenziali rischi potranno essere la mancanza di esperienza nello sviluppo e organizzazione del software, inoltre un potenziale rischio potrebbe essere la mancanza di ore di lavoro che ogni membro avrà da dedicare nel progetto. L'hardware non sarà un potenziale rischio di questo progetto grazie alla leggerezza del software.

7. Personale

In momenti diversi, il progetto richiederà diverse quantità di personale, con competenze diverse. L'inizio, la durata, l'importo e la competenza delle categorie di personale sono elencati in questa voce.

Competenze del team:

Andrei Petrisor, Andrea Rusconi, Uriel Fumagalli, Antonio Radu: Undergraduate Junior Developer

8. Metodi e tecniche

In questa sezione vengono forniti i metodi e le tecniche da utilizzare durante l'ingegneria dei requisiti, la progettazione, l'implementazione e le prove. In genere, qui viene descritto anche il modo in cui viene gestito il controllo della versione e della configurazione per i componenti software. Gran parte della documentazione tecnica sarà prodotta durante queste fasi. Si deve quindi precisare come sarà curata questa documentazione.

Viene descritto l'ambiente di prova e le apparecchiature di prova necessari. Durante il test, viene normalmente esercitata una pressione considerevole sull'apparecchiatura di test. Pertanto, questa attività deve essere pianificata con attenzione. L'ordine in cui i componenti vengono integrati e testati deve essere indicato esplicitamente. Inoltre, devono essere indicate le procedure da seguire durante i test di accettazione, ovvero i test sotto la supervisione dell'utente. I test sono discussi nel Capitolo 13.

TODO: rifinire quando stiamo scrivendo i test

9. Garanzia di qualità

Quale organizzazione e procedure verranno utilizzate per garantire che il software in fase di sviluppo soddisfi i requisiti di qualità dichiarati? I molti aspetti di un Piano di Assicurazione della Qualità possono anche essere trattati in un documento separato. Il tema della garanzia della qualità è discusso nel Capitolo 6 .

Riguardo ai fattori di qualità facciamo riferimento al modello di qualità esterno ed interno della standard iso 9126. Per quanto riguarda le organizzazioni e procedure per garantire i fattori di qualità si intende seguire il modus operandis delle organizzazioni SQA: monitorare adeguatamente il software durante il suo processo di sviluppo, assicurare il pieno rispetto degli standard e garantire che le inadeguatezze siano evidenziate affinché vengano sistemate dai developer.

10. Pacchetti di lavoro (workpackages)

I progetti più grandi devono essere suddivisi in attività, parti gestibili che possono essere allocate ai singoli membri del team. Ciascuna di queste attività deve essere identificata nel piano di progetto. La scomposizione gerarchica del progetto è rappresentata in una struttura di scomposizione del lavoro (vedi anche Sezione 8.4).

TODO: dividerci i compiti e determinare chi fa cosa man mano

11. Risorse

Durante il progetto sono necessarie molte risorse. L'hardware, i cicli della CPU e gli strumenti necessari per supportare il progetto sono elencati in questa voce. Occorre inoltre indicare il personale necessario per le varie fasi del processo.

Qualsiasi dispositivo con accesso ad internet può giocare.
Fortemente consigliato l'utilizzo da Computer/Laptop.

12. Budget e programma

Il budget totale per il progetto deve essere assegnato alle varie attività come indicato nella struttura di ripartizione del lavoro. Anche le attività devono essere programmate in tempo, ad esempio utilizzando un grafico PERT (vedi Sezione 8.4). In questa rubrica è indicato anche il modo in cui vengono tracciate le risorse e le altre spese. Il tema della stima dei costi e dei tempi è trattato ampiamente nel Capitolo 7 .

Il budget per questo progetto non sarà monetario ma definito da un quantitativo minimo di tempo utilizzato per lo sviluppo. Questo budget è identificato in 40 ore di lavoro minime per sviluppatore.

13. Cambiamenti

È stato affermato in precedenza che i cambiamenti sono inevitabili. Bisogna garantire che questi cambiamenti siano gestiti in modo ordinato. Sono quindi necessarie procedure chiare su come verranno gestite le modifiche proposte. Se il processo è agile, ogni iterazione comporta modifiche e queste vengono gestite in modo leggero. In realtà, non sono visti come cambiamenti. Nei processi più pesanti, ogni modifica proposta deve essere registrata e rivista. Quando una richiesta di modifica è stata approvata, è necessario stimarne l'impatto (costo). Infine, la modifica deve essere incorporata nel progetto. Le modifiche che vengono immesse tramite la porta sul retro portano a codice strutturato male, documentazione inadeguata e superamento di costi e tempi. Poiché le modifiche portano a versioni diverse sia della documentazione che del codice, le procedure da seguire per gestire tali modifiche vengono spesso gestite nel contesto di un piano di controllo della configurazione.

Utilizzando uno sviluppo AGILE del software la necessità di modifiche al progetto verrà gestita solo attraverso una riunione da parte di tutti i membri del team.

14. Consegna

Devono essere indicate le procedure da seguire per la consegna dell'impianto al cliente.

La procedura di consegna per questo progetto é stata definita con una data specifica (5 giorni prima della consegna della presentazione) e tramite il caricamento su un repository github di una versione eseguibile del progetto, insieme a tutta la documentazione e tutti i diagrammi utilizzati per lo sviluppo del progetto.