



Sistemi Informativi

Marino Segnan

Contesto

- A livello generale realizzare SW presenta gli stessi problemi organizzativi di una qualunque attività coinvolgente gruppi di persone.
- Esistono poi delle specificità del processo di sviluppo SW che hanno portato all'elaborazione di metodologie specifiche ad esso

Contenuti

- Accenneremo prima alle specificità di un progetto SW
- Considereremo successivamente un progetto SW come un progetto generico (relazioni tra persone e gruppi)
- Esamineremo infine un paio di metodologie specifiche di gestione di un progetto SW

Richiamo: complessita' di un SI

- Costruisco un SI per archiviare dei dati, es ricette di cucina, ad uso **personale**, su **file**.
- Vision:
 - tutti implicito, nella mia testa
- Dati:
 - nessuna “Classe”; solo stringhe
- Casi d'uso
 - unico: uso editor per modificare e cercare dati

Complessita' di un SI

- Mancano praticamente tutti i costi e vincoli:
- Il sistema e' su solo quando mi serve
- Sono nulli:
 - Tempo e costo di sviluppo, apprendimento, modifiche, coordinamento test etc
 - Il file e' portabile su tutti gli OS, in locale o remoto

Aumento complessita' di un SI

- Voglio formattare i campi per ricerche piu' sofisticate
- Uso editor piu' sofisticato, spreadsheet, interfaccia grafica DB:
- Introduco tempo di : apprendimento, costruzione interfaccia per DB, (tuttora costo senza altro impatto)
- Introduco potenziale costo di acquisto, problemi di **portabilita'** del tool e del DB
- Ancora da solo, forse nessun tecnico

Aumento complessita' di un SI

- Sempre da solo il programma e' per la zia cuoca, gratis (OpenSource) ma potrebbe chiedermi:
 - quando sara' pronto;
 - istruzioni d'uso (ancora a voce)
 - di correggere errori
 - di aggiungere funzionalita'
- E cosi' via ...
- Così la cordata degli sviluppatori ascende la montagna della complessita', confidando sulla conoscenza del terreno e sperando nel bel tempo

Aumento complessita' di un SI

- Dimensioni SW:
 - Numero/dimensione/complessita' componenti
 - Disponibilita'/evolvibilita'
- Requisiti committente
 - Riassumibili in costo/tempo
- Risorse umane
 - Overhead comunicazione/coordinazione
 - Gruppi con interessi intra-inter divergenti, spesso implicitamente!

Alcune definizioni

- Ciclo di vita del software
 - Un insieme di attività e le loro rispettive relazioni per sviluppare lo sviluppo di un sistema software
- Metodologia di sviluppo software
 - Una collezione di tecniche per costruire modelli applicati durante il ciclo di vita del software
- Software Engineering Process
 - Un SEP e' un processo che descrive come i requisiti dell'utente sono trasformati in un sistema software

Sviluppo software basato su componenti

- comprare
 - Componenti commerciali, Components Off The Shelf, Componenti riutilizzabili
- costruire
 - sviluppo ad hoc, sviluppo “from scratch”
- confrontiamo i due approcci

Componenti commerciali

- Vantaggi
 - Costi di licenza prevedibili
 - Tecnologia largamente usata e matura
 - Disponibile immediatamente
 - Struttura di supporto dedicata
 - Spesso indipendente da SW/HW
 - Ricchi in funzionalità

Componenti commerciali

- Svantaggi
 - Aggiornamenti frequenti
 - pagamento licenza prima dell'uso (*) e ricorrenti
 - dipendenza dal venditore
 - penalizzazione dell'efficienza runtime (a volte)
 - vincoli sulla funzionalità
 - integrazione a volte complessa
 - poco controllo su aggiornamenti e manutenzione
 - funzionalità non richieste consumano risorse aggiuntive
 - a volte inadeguate affidabilità e stabilità
 - Incompatibilità tra fornitori

(*) a volte “pay per use”

Componenti su misura (custom)

- Vantaggi
 - completa liberta' di cambiamento
 - implementazioni spesso piu' piccole e piu' semplici
 - spesso miglior prestazioni
 - controllo sullo sviluppo e i miglioramenti
- NB, alcuni componenti si acquistano quasi sempre:
 - DB relazionale
 - Window system
- mentre si costruisce il Controllo

Che soluzione adottare?

Scegliere che sistema fare puo' essere molto complesso e dall'inizio del corso stiamo dicendo che una causa di fallimento e' l'incorretta formulazione dei requisiti

<http://www.scribd.com/doc/39824816/Best-Practices-for-Software-Selection>

Suggerimenti

- Compatibilita' strategica
 - Avere un partner delle dimensioni adatte , altrimenti soluzioni troppo onerose
- Non farsi deviare dal proprio obiettivo a causa di demo o features addizionali ma inutili
- Aiutare il venditore a fare la miglior proposta
- Coinvolgimento del venditore
 - Dare tutte le informazioni possibili
- Non delegare completamente le scelte a chi ha gia' in passato fatto la valutazione del SW

Caratteristiche del Processo SW (1)

- Visibilità
 - E' esternamente visibile il progredire del processo ?
- Supportabilità
 - Può il processo essere supportato da strumenti (CASE).
- Affidabilità
 - Si possono scoprire errori del processo prima che risultino in errori del prodotto
- Robustezza
 - Può il processo continuare a dispetto di problemi inaspettati

Caratteristiche del processo SW (2)

- Mantenibilità
 - Può il processo evolvere per riflettere il cambiare della organizzazione o miglioramento del processo stesso
- Rapidità
 - Quanto velocemente può essere prodotto un sistema da una data specifica

Modello generico del processo in ingegneria

- Specifica - stabilire i requisiti e le restrizioni del sistema
- Progettazione - produrre un modello su carta del sistema
- Costruzione del sistema
- Test - valutare se il sistema soddisfa i requisiti
- Installare - consegnare il sistema al cliente ed assicurarne la operatività
- Mantenimento - riparare problemi che vengono incontrati

Costruire un ponte

– progetto

- prevedibile, preciso, disegni dettagliati,
- poche persone
- costo percentuale ridotto

– costruzione

- prevedibile in modo, tempo, costo
- costo percentuale elevato

Costruire del SW

- progetto (modello)
 - poco prevedibile,
 - uso di UML che e' poco testabile
 - molte persone
 - costo percentuale elevatissimo
- costruzione
 - quasi automatica, poco costosa (compilazione, configurazione, installazione)

Differenze rispetto al processo di costruzione del software

- Normalmente le specifiche sono incomplete/anomale
- C'è una distinzione non chiara fra specifiche, progettazione e costruzione (implementazione)
- Non c'è una realizzazione fisica
- Il software non si consuma - mantenimento non vuol dire sostituzione di componenti, ma generalmente aggiunta di requisiti
- la ripetitività è molto bassa

Fragilita' dei requisiti

– Fragilita' del SW

- piccolo errore, grandi ripercussioni
- Riprogettazione dei componenti difettosi

– Imprevedibilita' dei requisiti

- SW sembra facilmente modificabile
- difficile stimare costo requisiti
- difficile ottenere requisiti stabili dal cliente
- finche' non si vede non si puo' valutare
- il mondo attorno cambia imprevedibilmente

Modelli adottati nel SW

- Piu' tradizionali, burocratici:
 - a cascata (waterfall)
 - a spirale
 - Unified Process (UP) (slide ad hoc)
- Piu' recenti, **Agili**:
 - Extreme programming (XP)
 - Scrum
 - Kanban