

Evoluzione del software e gestione dei rischi

Lezione 13 di Sicurezza dei sistemi informatici 1

Docente: Giuseppe Scollo

Università di Catania, sede di Comiso (RG)
Facoltà di Scienze Matematiche, Fisiche e Naturali
Corso di Studi in Informatica applicata, AA 2006-7

Indice

1. Evoluzione del software e gestione dei rischi
2. gestione dei rischi: fasi e attività
3. valutazione dei rischi
4. controllo dei rischi
5. gestione delle configurazioni
6. derivazione di prodotti
7. controllo di versione
8. sistemi di controllo di versione
9. riferimenti

gestione dei rischi: fasi e attività

attività di gestione dei rischi nella produzione del software, secondo (Boehm, 1989), in due fasi:

1. **valutazione dei rischi**
 - identificazione dei rischi
 - analisi dei rischi
 - classificazione dei rischi

2. **controllo dei rischi**
 - pianificazione del controllo dei rischi
 - risoluzione dei rischi
 - rilevamento dei rischi

valutazione dei rischi

identificazione dei rischi, tecniche:

- liste di verifica (checklist)
- analisi dei criteri decisionali
- analisi delle premesse culturali
- supplemento di analisi

analisi dei rischi, scopo:

- valutare la **probabilità** che ciascuno dei rischi identificati si manifesti
- stimare la **perdita** causata in tal caso
- concetto: indice di esposizione al rischio **ER(e)** dell'evento **e**
$$ER(e) = P(e) * Perdita(e)$$

analisi dei rischi, tecniche:

- analisi delle decisioni
- analisi dei diagrammi PERT
- analisi what-if dei modelli di stima dei costi

classificazione dei rischi:

- in ordine di rilevanza o criticità rispetto agli obiettivi del progetto

controllo dei rischi

pianificazione del controllo dei rischi:

scopo: produrre un piano di intervento e di controllo per rilevare tempestivamente e fronteggiare l'eventuale manifestarsi dei rischi nel corso del progetto

concetto: coefficiente di riduzione del rischio $CRR(t,e)$

della tecnica t di riduzione del rischio per l'evento e

$$CRR(t,e) = (ER(e) - ER_{dopo}(t,e)) / C(t)$$

attività: analisi costi-benefici, riclassificazione dei rischi, integrazione nel piano operativo

risoluzione dei rischi: interventi previsti dal piano di controllo dei rischi, preventivi e in casu

rilevamento dei rischi:

attività di osservazione, misura e monitoring previste dal piano di controllo dei rischi

tecnica: hit list dei rischi

struttura organizzativa della gestione dei rischi:

responsabile

gruppo di controllo

gruppo di lavoro

gestione delle configurazioni

oggetto:

concetti, problemi, metodi e strumenti rilevanti alla **manutenzione** del software, ovvero alla sua **evoluzione** nel ciclo di vita

concetto di configurazione software (in prima approssimazione):

l'insieme dei moduli software e di informazione ad essi correlata (specifiche, documentazione di progetto, di collaudo, di utente, etc.) che permette di **costruire** una versione funzionante di un prodotto software, includendo in questa categoria anche le applicazioni Web

definizione di **gestione delle configurazioni** (in prima approssimazione, Babich (1986)):

insieme delle tecniche e delle metodologie che permettono di identificare, organizzare e controllare le modifiche apportate ad un programma sviluppato da un gruppo di programmatori, allo scopo di ottimizzare la produttività e ridurre il numero di errori

modifiche del software → conseguimento della massima soddisfazione di utente

coordinamento del lavoro di un gruppo di produttori

identificazione delle modifiche apportate

derivazione di prodotti

raramente la realizzazione di un prodotto software parte dal nulla: sempre più spesso risulta disponibile **codice riusabile** per fornire funzionalità diverse da quelle originarie, a fronte di limitate modifiche

rischio: importare nel prodotto eventuali difetti nascosti nel software preesistente

problemi:

tracciamento degli errori

determinazione delle dipendenze fra moduli nuovi e moduli riusati

completezza della manutenzione correttiva del software riusato

soluzione: si richiede esauriente documentazione della **derivazione del prodotto**

informazione che permette di ricostruire la **storia** della creazione ed evoluzione del prodotto

ad esempio (lista non esaustiva):

elenco delle configurazioni relative alle sue successive versioni, e per ciascuna di queste:

procedure utilizzate per creare gli oggetti derivati (moduli oggetto ed eseguibili)

autori delle creazioni

date di creazione

scopo della creazione

strumenti e documentazione di eventuali collaudi

controllo di versione

motivazioni: distinguere versioni diverse di uno stesso documento, modulo software o prodotto, risulta utile per due categorie di motivazioni affatto diverse:

distinguerlo in stadi diversi di sviluppo o manutenzione

distinguerne forme diverse in quanto adatte a contesti diversi

concetti di versione, due casi:

1. **revisione:** un modulo è una revisione di un altro modulo se lo sostituisce in tutte le nuove configurazioni create a partire dal momento in cui esso è reso disponibile; questo caso corrisponde all'evoluzione correttiva o perfezionativa del modulo
2. **variazione:** un modulo è una variazione di un altro modulo se, a partire dal momento della sua disponibilità, lo sostituisce solo in alcune delle nuove configurazioni, quelle cioè adatte a contesti in cui siano verificate le condizioni specifiche per le quali è realizzata la variazione

concetti correlati, rispettivamente:

1. **versioni concorrenti:**

meccanismi di identificazione univoca delle revisioni
memorizzazione incrementale delle revisioni
opportunità di documentare le modifiche
meccanismi di blocco (ingl. lock)

2. **famiglia di prodotti:**

insieme di prodotti ottenibili da configurazioni che differiscono solo per variazioni dei moduli

sistemi di controllo di versione

nello sviluppo concorrente, **problema:**

in assenza di un sistema di controllo di versione con un qualche meccanismo di blocco, l'accesso concorrente ad un'area condivisa, in cui autori diversi producono revisioni di documenti o di moduli software, può generare errori nel risultato o perdita di modifiche

soluzione: sandbox + lock ... (strict lock o soft lock?)

Revision Control System (RCS):

memorizzazione incrementale inversa
blocco per default
integrazione su richiesta esplicita

Concurrent Versions System (CVS):

gestione delle configurazioni in progetti software, repositories CVS in rete
integrazione per default
blocco su richiesta esplicita

Subversion (SVN):

controllo di versione esteso a directory, copie, ridenominazioni e meta-dati
controllo di versione di file di qualsiasi tipo, non solo testo, senza perdita di efficienza (algoritmo di analisi delle differenze di file binari)
e molto altro...

riferimenti

Boehm (1989) :

Tutorial: Software Risk Management
IEEE Computer Society Press, Washington D.C., USA

Babich (1986) :

Software Configuration Management: Coordination for Team Productivity
Addison-Wesley, Reading, MA, USA

RCS, Revision Control System

<http://www.cs.purdue.edu/homes/trinkle/RCS>

CVS, Concurrent Versions System

<http://ximbiot.com/cvs>

SVN, Subversion project

<http://subversion.tigris.org>

Scollo (2006) :

Evoluzione del software e gestione del cambiamento
Note del corso di Ingegneria del software 1, Corniso (RG)