

Come gestire il codice "legacy"

Un approccio efficiente per migliorare le basi di codice più datate

Fergus Bolger
Chief Technical Officer
PRQA

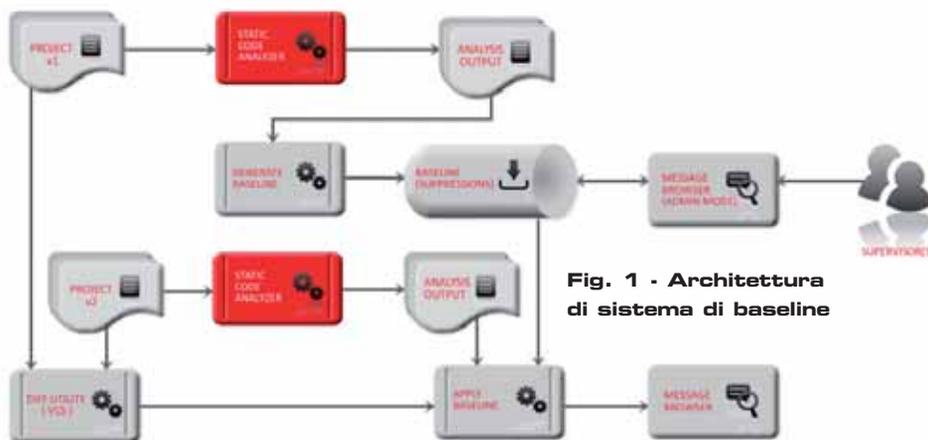


Fig. 1 - Architettura di sistema di baseline

Gli standard di codifica forniscono efficienti e concrete linee guida atte a individuare o prevenire errori e rendere il codice più leggibile, gestibile e sicuro. Per queste importanti attività, i tool QA•C e QA•C++ di PRQA sono tra gli strumenti di analisi più accurati e ad alte prestazioni al momento disponibili. Cosa succede quando si deve partire da una base di codice esistente che ha alle spalle una lunga storia di modifiche, di manutenzioni e di adattamenti?

Sono numerosi gli sviluppatori che devono lavorare con porzioni di codice di natura "ereditaria".

Benché sia sempre indispensabile prestare la massima attenzione ai bug messi in evidenza da QA•C e QA•C++, quando si adotta l'approccio orientato alla prevenzione, tipico di uno standard di codifica, verranno alla luce moltissime non-conformità e questo porterà rapidamente al proliferare di una serie di richieste di intervento il cui numero è decisamente superiore a ciò che si può ragionevolmente risolvere in un unico ciclo di manutenzione.

Spesso ciò che si può fare è migliorare selettivamente il codice per raggiungere in maniera graduale la piena conformità, eliminando progressivamente le procedure di codifica più pericolose e "sospette". Naturalmente questo non significa trascurare completamente le parti di codifica "peggiori": vi saran-

no sempre correzioni di bug critici che dovranno essere attuate immediatamente. Tuttavia, in un ambiente "legacy" vi possono essere frammenti di codice che rappresentano pericoli potenziali ma non immediati per i prodotti che li ospitano. In questi interventi di manutenzione del codice sarebbe auspicabile rimuovere tutte le non conformità della nuova codifica, senza intervenire sulla parte "legacy". In questo modo non si rischia perlomeno di peggiorare ulteriormente il codice.

Il "Legacy Mode"

PRQA ha sviluppato una tecnologia – denominata Legacy Mode - che rappresenta un valido ausilio per lo svolgimento di questa attività.

Questa modalità prevede che due versioni della stessa base di codice condividono le medesime problematiche di codifica - bug o non conformità agli standard di codifica: esse vengono dunque confrontate e abbinate quando coincidono. Nel momento in cui vengono applicate nuove tecnologie, è possibile mettere in evidenza solo le problematiche introdotte con le modifiche più recenti del codice. Queste rappresentano esattamente le correzioni che dovrebbe essere apportate al codice nel ciclo di sviluppo, prima del QA (Qualità Assurance), e del testing a livello di unità (ovvero del più piccolo componente software o modulo) e di quello integrazione

(relativo all'interazione tra i vari moduli). Per dimostrare le capacità e la versatilità applicativa, di seguito si analizza più in dettaglio questa soluzione tecnologica.

La tecnologia di analisi PRQA presente in QA•C e QA•C++, opera in primo luogo su ogni Translation Unit (TU). Con i risultati di questa analisi vengono ricavate delle diagnostiche sulle locazioni del codice sorgente e sui file header comuni. Queste diagnostiche sono di solito mappate su violazioni delle regole di codifica o - più semplicemente - su codici di bug specifici. Successivamente a questo primo esame vi è la fase di analisi Cross-Module (Cross- Module Analysis - CMA) che opera su tutta la serie di TU che costituiscono un'entità linkabile. Questa fase genera diagnostiche che permettono di individuare parecchi errori di temporizzazione del link e molte procedure errate.

Nell'analisi di questa raccolta di risultati diagnostici si evidenziano altri aspetti interessanti. Quando si combinano le uscite da ogni TU, è necessario rimuovere le diagnosi duplicate associate agli header comuni. Inoltre, le diagnostiche CMA sono abbinate a quelle della fase di analisi TU per ogni sorgente o header di file. Legacy Mode è una tecnologia baseline che deve tener conto di tutte queste fusioni, classificazioni e rimozione di duplicati al fine di elaborare una serie di diagnostiche "pulite".

L'analisi della baseline (specifica o prodotto formalmente revisionato che serve come base per ulteriori sviluppi) comporta un confronto dei set diagnostici tra due versioni della base di codice. Per comodità vengono indicate con il nome "nuova" e "vecchia" versione. Usando un'utility basata sulla differenziazione del testo è possibile ricalibrare la "nuova" posizione di ciascuna diagnostica nel punto in cui dovrebbe manifestarsi nella "vecchia" versione. Questo permette poi di rimuoverla temporaneamente, o sopprimerla, sulla base della sua presenza nella "vecchia" versione.

Tra le due versioni della base di codice, vi saranno ovviamente delle differenze che riguardano il nome dell'entità, delle posizioni di partenza delle funzioni o dei dati, delle lunghezze delle funzioni. Potrebbero inoltre essere stati aggiunti o rimossi degli oggetti. Un ulteriore elemento di complessità è la presenza di codice che può alterare il comportamento del pre-processor.

Una soluzione completa

La soluzione proposta da PRQA permette di affrontare l'insieme delle problematiche appena esposte. In sede di revisione dell'uscita diagnostica, la diagnostica precedente viene automaticamente soppressa. L'operazione di differenziazione del testo si verifica su ogni file header o sorgente della base di codice. Se non si vuole memorizzare la "vecchia" versione del codice, vi è la possibilità di utilizzare un sistema di controllo di versione (VCS) per gestire l'operazione di differenziazione. La

soluzione di base supporta i VVS più diffusi oltre a dare la possibilità di adattare rapidamente e facilmente altri VCS.

Il Message Browser è un componente GUI che rappresenta il mix di file sorgente/header riscontrabile nel mondo reale e offre una visualizzazione funzionale della diagnostica attiva e di base nascosta nel sorgente, con sommari organizzati per file, gerarchia del messaggio e fase di analisi.

Il componente Message Browser si distingue per la presenza di una nuova modalità di funzionamento denominata "baseline administration". L'utente può navigare all'interno dell'intero set di strumenti diagnostici di base e ri-attivare una qualsiasi selezione diagnostica al fine di farla riapparire in ogni analisi base di una "nuova" versione del sorgente. Un semplice strumento di navigazione permette di scegliere facilmente un singolo diagnostico o tutti quelli relativi a un determinato id di messaggio, gruppo di messaggi o livello di messaggio nella gerarchia. Il salvataggio delle modifiche al set di baseline permette di utilizzare il nuovo livello di riferimento da parte di tutto il team di sviluppatori, in modo che i progettisti possano sempre concentrarsi sugli aspetti diagnostici più importanti e lasciare da parte quelli marginali.

Il Legacy Mode mette a disposizione una funzionalità efficace per la gestione di basi di codice di generazione precedente e consente interventi mirati sui bug più importanti.

Il tutto permette di trasferire al team di sviluppo un set di diagnostica prioritario per le azioni immediate. Le basi possono essere ripristinate in cicli successivi di fix/analisi per i singoli o per i gruppi di sviluppatori.

Versatilità d'utilizzo

Questa tecnologia prevede alcune interessanti applicazioni diverse dal semplice confronto tra due versioni di codice sorgente. La si può infatti utilizzare per testare un'applicazione nell'ambito di host e target. Se si modifica la configurazione di analisi per indirizzare i set di tipi fondamentali di dimensioni ridotte e i loro allineamenti di byte è possibile modellare e controllare nuovi problemi che possono manifestarsi in termini di conversione e di allineamento del puntatore. Un altro caso d'uso potrebbe essere l'individuazione dei diagnostici che emergono da una versione più recente di QA•C o QA•C++ prima della distribuzione in produzione.

Questa forma estremamente raffinata di analisi differenziale è totalmente nuova ed è appannaggio esclusivo dei tool PRQA. Unitamente alle capacità di diagnostica e alla forte concentrazione sul linguaggio tipici del set di strumenti PRQA, la soluzione di base Legacy Mode rappresenta un mezzo efficace per aumentare la qualità del codice.

PRQA

www.programmingresearch.com