

La gestione della qualità del codice

Gestire la qualità del codice è un aspetto che è diventato sempre più importante per le applicazioni embedded a causa anche della maggiore quantità di software utilizzato

Francesco Ferrari



li attuali sistemi embedded ospitano una quantità di software notevolmente maggiore rispetto a qualche anno fa, così come è aumentata sensibilmente la complessità di questa parte dei progetti. Gli sviluppatori hanno quindi bisogno di metodologie e strumenti sempre più articolati per

assicurare lo progettazione dei prodotti in tempi che stanno diventando sempre più brevi per inseguire il time to market.

La gestione della qualità del software, che solitamente viene distinta fra qualità funzionale e qualità strutturale, è importante per diversi motivi che spaziano dalla gestione di rischi, per esempio che il software non funzioni correttamente, alla gestione dei costi, per esempio, con il riutilizzo del codice in progetti successivi.

In un scenario di questo tipo, la corretta gestione della qualità del software diventa quindi un aspetto molto importante per raggiungere gli obiettivi prefissati per soddisfare le richieste del mercato.

La qualità del codice

Il concetto di qualità del codice non fa quindi riferimento unicamente all'assenza di bug, ed è tutt'altro che indefinito, ma, anzi, esistono diversi specifici riferimenti normativi che ne definiscono con molta precisione i modelli e le caratteristiche e il contesto.

Oltre alle normative più generali legate alla qualità e alle certificazioni dei processi, ci sono, per esempio, le norme ISO/IEC 9126,25010, 12119 e 9241.

In generale, parlando di modelli di qualità del software, si fa riferimento a diversi livelli dove sono descritte le proprietà qualitative, solitamente astratte, tramite una serie di caratteristiche che invece sono perfettamente misurabili. Il grado di possesso di queste sottocategorie che ha il codice può essere così valu-

tato rispetto a una scala di riferimento. Per esempio, in questo ambito, l'organizzazione ISO ha prodotto, diverso tempo fa, alcune norme che definiscono il modello di qualità del software in diversi contesti di utilizzo. La ISO/IEC 9126-1 definisce sei caratteristiche di qualità principali a loro volta suddivise in



27 sottocaratteristiche misurabili tramite delle metriche raccolte in tre technical report (ISO/IEC 9126-2, 3 e 4). Le specifiche ISO/IEC 9241, invece sono riferite alle caratteristiche di qualità di un software definito "usabile", mentre quelle ISO 12119 definiscono le caratteristiche di qualità di un software "Commercial off the shelf" (COTS).

La norma 9126-1 definisce la qualità del software come l'insieme delle caratteristiche che incidono sulla capacità del prodotto di soddisfare requisiti espliciti o impliciti. Il modello proposto si articola su quattro livelli. Nel primo sono presenti tre punti di vista (esterno, interno ed in uso) mentre il secondo precisa i principali attributi che il codice deve avere in base ai tre punti di vista.

Gli altri due punti specificano le sottocaratteristiche e le metriche per realizzare le misure. Il processo di valutazione è definito nella norma ISO 14598.

Per avere un'idea della complessità di queste specifiche, basta notare che le caratteristiche che definiscono la qualità esterna di un software sono la funzionalità, l'affidabilità, l'usabilità, l'efficienza, la manutenibilità e la portabilità, a

loro volta suddivise in 27 sottocaratteristiche rigidamente definite.

Uno dei limiti delle norme ISO 9126 però è che non permettono di separare nettamente le varie caratteristiche dai loro effetti sul software.

I problemi e le soluzioni

Le problematiche relative allo sviluppo del codice e alla gestione della sua qualità sono molteplici, a seconda del contesto, e partono dalla necessità di rilevare ed eliminare i bug, al coordinamento dei team di sviluppo, anche esterni, visto che non è più pensabile per la maggior parte dei progetti che ci possa essere unico sviluppatore o un solo piccolo team, per arrivare alla necessità di documentare con estrema precisione le varie fasi dello sviluppo per poter intervenire successivamente in modo

La definizione e la manutenzione della qualità necessitano di un processo continuo

tempestivo ed efficace con modifiche, upgrade o riutilizzo del codice.

Non deve essere trascurata, infatti, l'ottimizzazione del codice di prodotti già rilasciati attraverso aggiornamenti, ma, ovviamente, deve essere anche evitato che siano rilasciati nuovi prodotti finché non si raggiunge il livello di qualità desiderato.

In generale, gli esperti ribadiscono che è molto più economico, sul versante dei costi di sviluppo, ma non solo, trovare un bug prima della fase di test, e, come è

facile intuire, lo è molto di più trovarlo prima che il prodotto sia distribuito sul mercato. Questo spiega perché per i team di sviluppo sia prioritario trovare e correggere i bug il prima possibile durante il ciclo di sviluppo.

Per quanto riguarda le possibili soluzioni, la definizione e la manutenzione della qualità del codice necessitano di un processo continuo, che dura per tutto lo sviluppo dell'applicazione. Normalmente la qualità del codice è un aspetto che viene fissato in fase di pianificazione e controllata tramite una serie di test e revisioni del codice.

La gestione della qualità del codice per la ricerca di potenziali problemi implica solitamente l'applicazione di diverse metodologie, come per esempio la revisione paritaria del codice, l'analisi statica del codice sorgente, misurazioni e automazione dei test. La revisione paritaria del codice (peer review) per esempio è considerata da molti come una delle strade più economiche e efficaci per migliorare la qualità. In questa fase gli sviluppatori studiano a vicenda codice sorgente, evidenziando i potenziali problemi già in fase di sviluppo. Un controllo da parte di altri sviluppatori diventa possibile per trovare molti problemi già nelle fasi iniziali dello sviluppo del codice, quando cioè è molto veloce ed economico risolverli.

Altre strategie prevedono, invece, l'adeguamento a standard che prevedono l'applicazione di "best practices". I suggerimenti che si possono trovare sono numerosi. Per esempio, la gestione della qualità del codice può trarre vantaggi da operazioni come la preparazione in anticipo della classificazione dei rischi per non far perdere tempo successivamente al team di sviluppo. Analogamente può essere utile stabilire un ordine di priorità del codice da sottoporre a verifiche più dettagliate. Microsoft, per esempio, sul suo sito offre numerosi spunti per gestire la qualità del codice, sono suggerimenti interessanti anche per la parte legata alla sicurezza.

Le strategie attuabili comunque sono numerose, ma in molti casi un aiuto importante deriva dalla disponibilità di ambienti di sviluppo idonei, che si stanno diffondendo sempre di più. Analogamente ci sono aziende che, oltre a specifici prodotti, hanno online diversi white paper che consentono di inquadrare in modo preciso le problematiche relative alla qualità del codice.

