



# Accelera la compatibilità agli standard di sicurezza per C & C++ con il test di unità e di integrazione automatizzato di Cantata



Guidare la qualità del software embedded



## Caratteristiche di Cantata in evidenza

- > Test Harness automatizzato e creazione di test case
- > Ampio supporto di piattaforme e integrazione di toolchain
- > Interfaccia Utente Eclipse di facile uso e test scritti in C/C++
- > Tracciabilità bidirezionale dei requisiti
- > Straordinario controllo delle chiamate di funzione per simularle ed intercettarle
- > Inserimento flessibile del codice utente
- > Supporto per il Test Driven Development
- > Analisi di copertura (code coverage) integrata
- > Test di regressione automatizzato
- > Verifica dei test per le modifiche al codice
- > Tool certification kit gratuito per i maggiori standard di sicurezza

## Tipi di testing supportati

- black-box / white-box
- positivo / negativo
- requisiti / robustezza
- set di dati in ingresso singolo/grande
- procedurale / object oriented
- dal codice / dalle intestazioni
- simulazione/intercettazione delle chiamate
- isolamento / integrazione
- esecuzione su host / su target
- di nuovo genere / di regressione

## Cantata funziona con il vostro ambiente

Cantata si installa su sistemi operativi Windows® e Linux®, con un IDE Eclipse® integrato, o con un set di plug-in per Eclipse®. Aggancia le catene di strumenti di compilazione esistenti per creare automaticamente progetti di test integrato con un ampio set di toolchain di sviluppo embedded:

- ✓ IDEs / RTOSs
- ✓ GCC, Microsoft, Cross-compilatori
- ✓ Debuggers
- ✓ Build / Integrazione continua
- ✓ Gestione della configurazione software
- ✓ Gestione del requisiti

Le impostazioni di compilazione delle toolchain non-Eclipse sono importate automaticamente nei progetti di test di Cantata. Per avere conferma di come i vostri strumenti e le vostre piattaforme siano supportati, vi invitiamo a contattare QA Systems.

## Utilizzo illimitato di target embedded

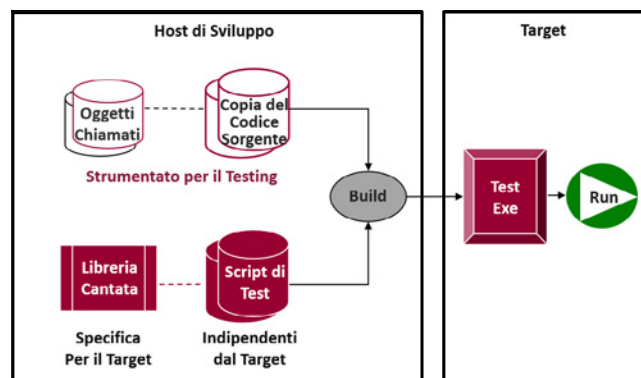
Per gli ambienti su target, tramite un wizard incorporato sono disponibili per l'uso, senza alcuna restrizione di licenza, tutti i deployment. I deployment di Cantata consistono di librerie ed opzioni di configurazione compatibili a livello binario con il codice. Gli stessi sono testati e verificati per l'uso certificato su progetti relativi alla sicurezza.

## Testing agevole e flessibile su target

I test di Cantata (script di test in C/C++ platform independent e deployment specifici per i target embedded) vengono generati come eseguibili C/C++, scaricati ed eseguiti sulla piattaforma target proprio come si farebbe con il proprio codice, mediante:

- ✓ Simulatore
- ✓ Emulatore
- ✓ Target Board

I risultati dei test funzionali e di copertura del codice sono diretti all'host per la diagnostica e la reportistica. Il processo è completamente automatizzato utilizzando i Makefile di Cantata, gli script di test e le personalizzazioni di piattaforma per un test facile e flessibile su target dalle interfacce utente o a linee di comando.



Il codice utente è gestito tramite script di test portatili con le librerie compatibili con il target, e costruito come un unico file eseguibile di test per andare in esecuzione su molteplici piattaforme host o target. La strumentazione viene utilizzata per l'accesso white-box e la copertura, per cui il codice di produzione non viene mai modificato per il testing.

# Perché I leader di settore utilizzano Cantata



## Taglio dei costi di conformità agli standard

Cantata soddisfa i requisiti di test dinamici delle norme di sicurezza del software. È una soluzione unica per il test dinamico di unità e di integrazione su host e piattaforme target.

La certificazione dei tool di sviluppo può essere un pesante fardello dei costi di conformità. Cantata è stato certificato in maniera indipendente da SGS-TÜV SAAR GmbH, e fornisce gratuitamente un tool certification kit con tutto l'occorrente necessario.

Capire come rispettare le norme di sicurezza del software è complesso e richiede tempo. Una guida completa sull'uso delle potenti funzioni di Cantata per soddisfare ogni requisito di test dinamico dello specifico standard, è fornita anch'essa gratuitamente. Questa dotazione e la combinazione della guida con le funzionalità dello strumento permette con Cantata di dimostrare più rapidamente e a minori costi la conformità agli standard.

Certificato per :

- > ISO 26262:2018 (Automotive)
- > EN 50128:2011/A2:2020 (Ferroviario)
- > EN 50657:2017 (Materiale Rotabile Ferroviario)
- > IEC 62304:2006 (Medicale)
- > IEC 61508:2010 (Industriale)
- > IEC 60880:2006 (Nucleare)

Qualificabile per:

- > DO-178C / DO-330
- > Altri standard richiesti

## Ridurre il rischio di difetti del software

Il richiamo di prodotti e l'infezione di una vasta reputazione aziendale e del marchio possono superare di gran lunga i costi di sviluppo delle singole applicazioni. Il test di Unità è il modo più completo per testare il codice dell'applicazione e prevenire i bug nei dispositivi che vengono distribuiti.

Gli slittamenti del progetto possono essere attenuati spostando lo sforzo per il lavoro di verifica nelle prime fasi del ciclo di vita di sviluppo del software. Questo riduce i rischi di ritardi nel corso delle fasi di test successive, perché i componenti testati con il test di unità sono più facili e prevedibili da integrare.

L'idoneità allo scopo di cause contro aziende e privati è oggi un rischio sempre crescente. Dove le aziende non riescono a servirsi di pratiche di settore, ormai ampiamente accettate come un approfondito test di unità con Cantata, non possono avvalersi della difesa giudiziaria dello „stato dell'arte“ contro tale contenzioso.

## Minori costi di testing

Effettuare il testing in anticipo riduce i costi, minimizzando il lavoro di ri-codifica in fasi successive dello sviluppo. Gli sviluppatori possono identificare i difetti con il test di unità e di integrazione di Cantata non appena ogni componente è disponibile.

L'elevato costo per la conformità agli standard del test di unità e integrazione può essere drasticamente ridotto attraverso l'automazione. I requisiti di testing dinamico per le norme di sicurezza vengono raggiunti e rispettati in modo più veloce e conveniente con l'automazione in Cantata di:

- > Generazione di Test Framework
- > Generazione di Test Case
- > Test di base sul codice legacy/esistente
- > Esecuzione di Test su Host o su Target
- > Test di regressione in integrazione continua
- > Generazione di Report e Diagnostiche

L'integrazione di strumenti in una toolchain può aggiungere costi di testing nascosti. La stretta integrazione di Cantata con gli ambienti di cross-compilazione, ed i suoi test intuitivi in codice C / C ++ in Eclipse® o editor di codice, lo rendono di facile inserimento in qualsiasi toolchain. E ,più veloce da apprendere e più conveniente da utilizzare, come lo vogliono gli sviluppatori.

## Ridurre i tempi di mercato

I leader di settore riconoscono la necessità di spedire velocemente senza mettere a rischio la qualità. I Test di Cantata offrono due vantaggi chiave in termini di tempo per i manager dello sviluppo:

- > La collaborazione e l'efficienza del team è migliorata con corrispondenti test strutturati e report pronti per la certificazione.
- > I tempi di Integrazione sono più brevi e più prevedibili quando si integrano componenti software collaudati singolarmente.

## Framework di test flessibile

Una struttura di test flessibile (script di test e una libreria di supporto) consente qualsiasi combinazione di stili di test per entrambi test di unità e test di integrazione scalabile. I test possono essere rivisti in una GUI o direttamente come codice C / C ++, e vengono mandati in funzione come eseguibili sia su host che su diverse piattaforme target.

## Test Driven Development (TDD)

Cantata consente di scrivere i test non appena vengono creati prototipi di funzione all'interno dei file di intestazione. Ciò consente di costruire il framework di test prima che il corpo del software in prova sia completamente implementato. Cantata per TDD migliora le tradizionali tecniche TDD black-box, dando accesso a test white-box completi su interni di codice incapsulati, come dati e funzioni privati / statici.

## Black and white box testing

La Creazione altamente automatizzata di test case offre potenza e precisione per il test black-box ed il test white-box maggiormente efficiente ed approfondito. Il potente test black-box è attivato con un loop di test parametrizzati selezionabili dall'utente o predefiniti. Un effettivo calcolatore combinatorio e l'import / export CSV per grandi insiemi di dati.

Il preciso test white-box attraverso l'strumentazione di Cantata accede automaticamente al codice incapsulato direttamente dallo script di test, senza compilazione condizionale, dando controllo su funzioni e dati statici e privati.

## Test di robustezza

Il test di robustezza è reso facile con il set di regole di Cantata dei valori predefiniti per i tipi di dati di base, in un loop di Test Case. Tutti i dati globali accessibili vengono controllati automaticamente per modifiche accidentali.

## Testing object oriented

I test in stile object oriented di Cantata sono implementati come classi per i metodi di test, modelli o gruppi di classi. Includono funzioni automatizzate di:

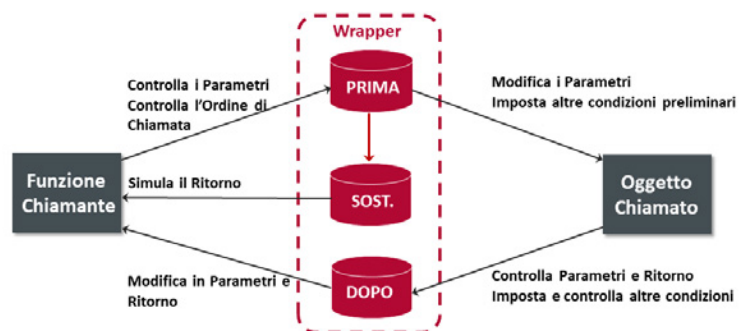
- > Riutilizzo dei test case attraverso una gerarchia di ereditarietà parallela.
- > Struttura di eredità delle classi di test per classi ereditate.
- > Attuazione concreta delle classi astratte di base (ABC) o metodi puramente virtuali.
- > Risoluzione delle dipendenze da riferimenti indefiniti che non sono chiamati direttamente dal codice

## Esclusivo controllo delle chiamate

Cantata genera automaticamente i controlli di test sia per simulare (stub) e intercettare (wrap) tutte le chiamate di funzione dal software in prova, fornendo:

- > Controlli automatici opzionali su parametri e dati
- > Molteplici istanze per comportamento di chiamata differenziato
- > Flessibile verifica dell'ordine di chiamata in ogni test case
- > Interfaccia di rilevamento errori e di inserimento errori
- > Test di accoppiamento

I Wrapper intercettano le chiamate per verificare il reale, non il presunto, comportamento simulato dell'oggetto chiamato. Dove la simulazione non è possibile o auspicabile (chiamate interne all'integrazione, chiamate del sistema operativo, interfacce hardware, ecc), i wrappers forniscono un potente unico controllo delle chiamate a Cantata.



## Traccia dei requisiti

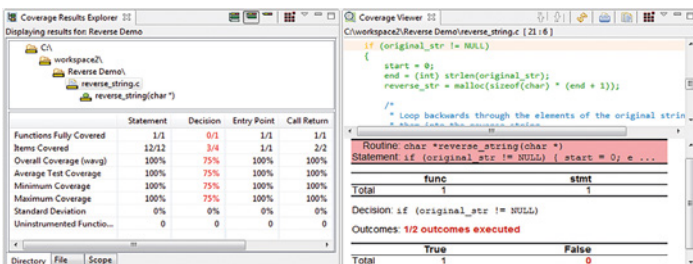
La tracciabilità bidirezionale di requisiti o piani di test è richiesta da tutte le norme relative alla sicurezza. I requisiti o i set di piani di test vengono importati in un server Cantata, i rapporti vengono assegnati con una interfaccia drag-and drop, e poi esportati con lo stato dei risultati dei test di Cantata e le informazioni sulla copertura del codice.



L'importazione / esportazione può avvenire tramite CSV, Microsoft Excel® o formato di interscambio di requisiti (ReqIF™), per adattarsi al flusso di lavoro e allo specifico strumento di gestione dei requisiti (ad es. IBM® Rational® DOORS®, PTC Integrity®, Polarion® REQUIREMENTS™, Intland codeBeamer®).

## Copertura del Codice

La copertura di codice in Cantata fornisce misurazione oggettiva di quanto a fondo i test hanno eseguito il codice sorgente (guidato o meno dai test di Cantata). I set di regole di copertura di Cantata specifici per ogni Standard lo rendono facile da usare, automatizzando la strumentazione, la presentazione dei dati e il controllo integrato dei livelli di copertura del codice richiesti.



La copertura del codice é misurata utilizzando I seguenti parametri:

- › Punti di Entrata
- › Ritorni di Chiamata
- › Statements
- › Dichiarazioni e Blocchi di Base
- › Decisioni (rami)
- › Condizioni
- › MC/DC
- › Loops
- › Operatori relazionali

“Cantata Build Variant Coverage” raccoglie i dati sul codice sorgente, quando le definizioni vengono utilizzate per creare codice eseguibile in diverse varianti, fornendo analisi e report certificati della copertura aggregata su tutte le varianti.

Le diagnostiche pin-point possono filtrare o aggregare la copertura su tutto il codice del progetto, fare drill-down sui costrutti individuali del codice, all’interno di ogni riga, per test case, per esecuzione di test, e tipo di metrica e contesto (eredità, threads, stati, accoppiamento dei dati etc.). L’ottimizzazione automatica dei test case aiuta la selezione dei vettori dei test case da un ampio set di dati e riduce lo sforzo complessivo per il test di regressione.

## Test Continuo



I Makefile Cantata vengono generati automaticamente alla creazione del test per compilare, collegare, distribuire, eseguire e recuperare i risultati delle suite di test in modalità batch. Possono essere utilizzati con i makefile esistenti, e sono facilmente integrabili con strumenti di integrazione continua open source o commerciali per il test di regressione, quali Jenkins e Bamboo.

## AutoTest

La generazione automatica di vettori di test case può esercitare una copertura del 100% del codice mentre si controllano i dati, i parametri e l’ordine di chiamata. Questo può colmare le lacune di copertura del codice o creare test unitari completi facilmente collegati ai requisiti, riducendo così la dipendenza da test unitari generati manualmente e da costosi test di sistema.

## Testing Basato sulle Modifiche al Codice

L’analisi della modifica del codice di Cantata aiuta ad automatizzare la manutenzione dei test unitari. Vengono identificate le modifiche che incidono sui test esistenti e vengono suggeriti aggiornamenti adeguati. Gli script di test vengono quindi ‘refactored’ automaticamente.

## Diagnostica e report



Cantata fornisce una potente diagnostica filtrabile dei risultati dei test e della copertura del codice all’interno dell’interfaccia grafica di Eclipse® oltre a report flessibili, configurabili dall’utente in XML e in testo ASCII e HTML pronti per la certificazione.

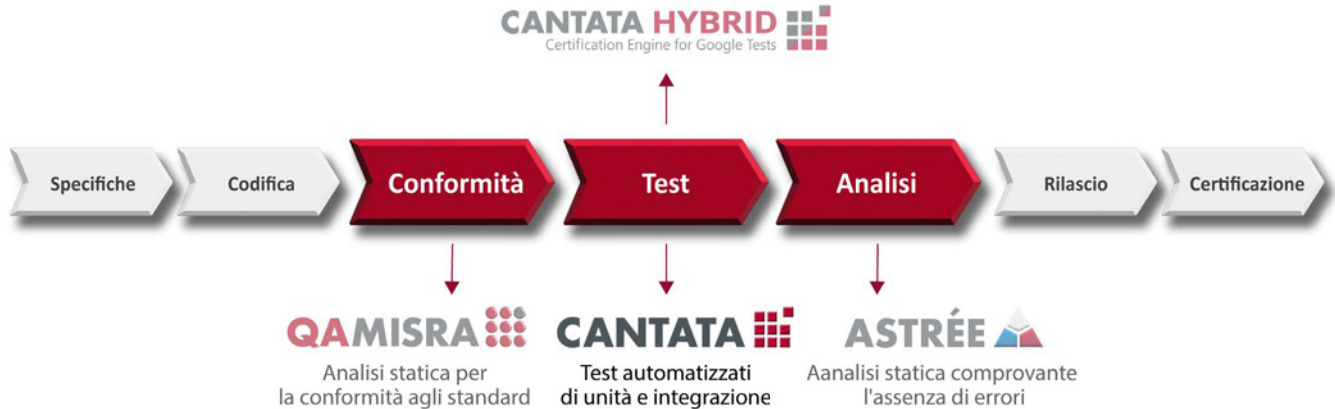
## Add-on opzionale : Team Reporting

Il prodotto addizionale Cantata Team Reporting, con una architettura client-server, interfaccia web e REST API, permette ai team distribuiti di lavorare più efficacemente. Vedere la scheda tecnica di Team Reporting per maggiori dettagli.



# Strumenti incentrati sulla verifica

Gli strumenti di analisi statica e di test dinamico del software di QA Systems favoriscono la verifica nel flusso lineare di sviluppo del software di seguito riportato. Si consiglia di applicare un approccio sequenziale a queste fasi di verifica con strumenti mirati per ogni scopo



**Conformità** Usa **QA-MISRA** per ottenere la conformità agli standard di codifica direttamente sul desktop dello sviluppatore.

**Test** Usa **CANTATA** per l'esecuzione dinamica automatizzata del software standard compliant.  
Usa **Cantata Hybrid** per generare risultati di test Cantata certificati da test Google esistenti.

**Analizza** Usa **ASTRÉE** per provare l'assenza di errori di run-time nell'intera applicazione.

NB: ASTRÉE utilizza la stessa configurazione di QA-MISRA, quindi lo sforzo per applicarlo successivamente a un progetto QA-MISRA è minimo.

## Opzione bundle su speciale licenza condivisa

QA-MISRA e CANTATA condividono la stessa tecnologia di licenza. Ciò consente ai clienti di ottenere una soluzione in bundle per entrambi gli strumenti, per condividere contemporaneamente lo stesso pool di licenze e per integrare gli strumenti nell'IDE basato su Eclipse.

Quando QA-MISRA viene acquistato in bundle con CANTATA o quando una licenza CANTATA esistente viene convertita in bundle, sono disponibili prezzi combinati molto interessanti. Contattateci per maggiori informazioni.

### Prova gratuita

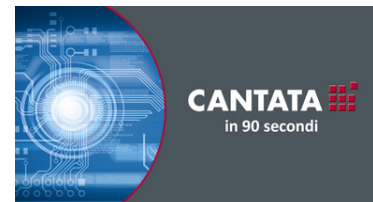
Provate Cantata con il vostro codice e il vostro ambiente di sviluppo.

 **START TRIAL**

### Cosa aspettare nella prova

- > Una demo personalizzata e una sessione di domande e risposte.
- > Installazione molto semplice.
- > Semplice configurazione per il compilatore.
- > Uso illimitato, licenza di prova limitata nel tempo.

### Scopri di più



[qa-systems.com/tools/cantata](http://qa-systems.com/tools/cantata)



Con uffici a **Stuttgart, Germania** | Bath, Regno Unito | Milano, Italia | Boston, Stati Uniti | Da Nang, Vietnam | Bengaluru, India  
[www.qa-systems.it](http://www.qa-systems.it) | [www.qa-systems.com](http://www.qa-systems.com)