



---

## LA SCHEDA

---

**Julia**, acronimo di **Java Universal Interpretation and Abstraction**, è un analizzatore di codice Java sviluppato, negli ultimi 10 anni, nell'ambito di progetti di ricerca coordinati da un team italo-francese con il coinvolgimento dell'Università di Verona e quella de La Réunion. Il team di ricerca accademico, che detiene il 35% del capitale sociale (il restante 65% è stato acquisito da Corvallis), è composto da **Fausto Spoto**, docente di informatica all'Università di Verona, **Roberto Giacobazzi**, ex preside della Facoltà di Scienze MM.FF.NN. dell'Università di Verona e **Frédéric Mesnard**, ex direttore del Dipartimento di matematica e informatica a La Réunion.

**Julia** è un vero e proprio **“cacciatore di bug”** in programmi Java e Android. Al mondo esistono poche decine di strumenti creati allo stesso scopo. Ciò che distingue Julia dagli altri competitor e che la rende un prodotto unico al mondo per livelli di precisione ed efficacia è l'utilizzo delle più moderne e riconosciute tecniche di **interpretazione astratta** che consentono un'analisi del software in **modalità statica e semantica**. Julia non richiede il codice sorgente del programma analizzato ma solo il bytecode compilato. Il metodo riduce il numero di stati possibili del sistema tramite l'astrazione e consente, grazie alla disponibilità di memoria e potenza di calcolo, di analizzare tutte le variabili del programma in ogni cammino di esecuzione. Il tool elenca una serie di warning (potenziali errori) che devono essere analizzati a cura del programmatore, che ha in definitiva la responsabilità di decidere se la segnalazione è effettivamente un problema e di correggere o irrobustire il programma. Questo metodo garantisce di trovare tutti i bug del tipo indagato.

I principali competitor sul mercato, invece, fanno largo uso della tecnica di pattern-matching per rilevare errori: semplificando molto, questa tecnica consiste nel cercare meccanicamente nei file sorgente “sequenze” riconosciute come tipiche fonti di errore. Così facendo, lo strumento si dimostra incapace di individuare tutti gli errori, ma solo quelli che rientrano nel pattern cercato. Inoltre, la necessità di disporre dei file sorgenti spesso impedisce di analizzare librerie di terze parti.

Nello sviluppo del Software utilizzare gli **SCA (Software Code Analyzer)** consente di velocizzare a vari livelli la diagnostica in fase di codifica, di test e nelle fasi successive legate alla gestione delle patch e degli aggiornamenti. La valenza degli SCA è strategica anche rispetto al tema della sicurezza in quanto gli hacker costruiscono i loro attacchi principalmente sui difetti e sui malfunzionamenti di un programma. Garantendo la qualità dello sviluppo, gli strumenti automatici di analisi non solo aumentano l'efficienza del processo di programmazione ma contribuiscono a garantire l'invulnerabilità del software e quindi la reputazione dell'azienda.

Anticipare l'uso di uno SCA nel **SLM (Software Lifecycle Management)**, consente dunque di snellire i processi, abbattere i costi di debugging e ottenere software più sicuri, funzionali, utilizzabili, disponibili, efficienti, manutenibili e portabili.

Il mercato del **Automated Software Quality, ASQ**, in cui Julia come SCA si inserisce, presenta un notevole gap tra la necessità delle aziende di migliorare la loro produttività e la mancanza di soluzioni in grado di garantirlo. Meno di 10 players principali hanno creato un mercato mondiale di oltre 2,4 miliardi di dollari, in continua crescita.