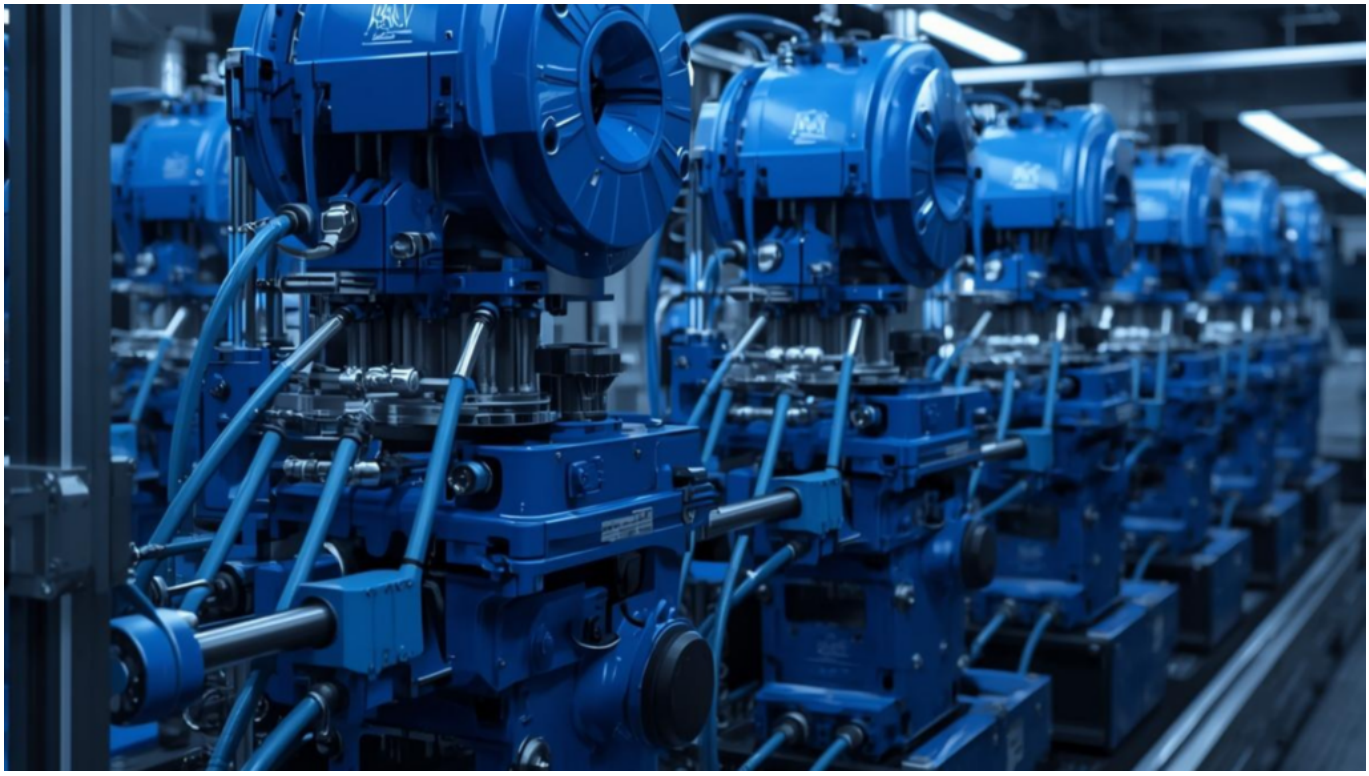
4.4 Sistemi di test interconnessi: dall’automazione locale all’orchestrazione dei dati con QX

Category: Insight
scritto da admin | Marzo 2, 2026



Nel precedente articolo abbiamo visto come adottare componenti COTS, piattaforme modulari e framework di sequenza riusabili permetta di progettare banchi prova più veloci da sviluppare e più semplici da manutenere. Il passo successivo, inevitabile quando i banchi aumentano di numero e si distribuiscono tra stabilimenti e laboratori diversi, è smettere di considerarli applicazioni isolate e iniziare a trattarli come nodi di un’unica infrastruttura digitale di Test & Measurement, dal banco al cloud.

Oggi un banco prova non esegue “solo” misure: utilizza strumenti condivisi, gestisce configurazioni complesse, deve garantire tracciabilità, dialoga con sistemi di produzione e qualità e genera grandi quantità di dati sensibili. Il valore non è più solo nelle funzionalità del singolo applicativo di test, ma nella capacità di far lavorare in modo coordinato decine di sistemi diversi, mantenendo coerenza, visibilità e controllo nel tempo.



Dai framework locali all’infrastruttura di test

Con l’introduzione di framework come SPOT e PATH abbiamo portato ordine dentro al singolo banco: sequenze strutturate, gestione uniforme degli errori, logging coerente, integrazione standard con l’hardware. Questo livello rimane fondamentale, ma da solo non basta più quando i sistemi di test diventano molti, distribuiti e critici per il processo aziendale.

Nella pratica quotidiana, vediamo spesso organizzazioni con 10, 20 o 50 banchi sviluppati nel corso degli anni in LabVIEW o Python, ognuno con stile, strutture dati e modalità di configurazione proprie. Anche quando il framework sottostante è stato introdotto, la gestione complessiva resta fragile se ogni postazione continua a vivere come una “isola”: chi è responsabile di che cosa, quale versione di sequenza gira dove, quali strumenti sono realmente installati e tarati, dove finiscono i risultati.

SPOT e PATH sono nati proprio per evitare che ogni sviluppatore “reinventi” il modo di fare test, ma danno il loro massimo quando sono inseriti in un disegno più ampio, che prevede un livello di orchestrazione superiore. È qui che entra in gioco QX come piattaforma di governo della flotta di sistemi di test.

Quando il problema non è più il test, ma la gestione

Finché i banchi sono pochi, molte complessità si mascherano dietro a procedure manuali e competenze individuali. È quando il numero di sistemi cresce, quando il test è distribuito su più siti o quando entrano in gioco requisiti di compliance più stringenti (audit, tracciabilità, validazioni) che emergono i problemi strutturali.

Situazioni tipiche che incontriamo sul campo: versioni di sequenze diverse installate su banchi teoricamente “identici”, parametri di test modificati localmente e non documentati, strumenti non più in stato di taratura che continuano a produrre dati considerati validi, risultati salvati su PC locali o cartelle condivise non governate. In questi contesti il collo di bottiglia non è più la capacità di eseguire test, ma la difficoltà di sapere con certezza che cosa è stato testato, come, con che cosa e dove sono i dati quando serve ricostruire la storia di un componente.

Senza un livello di orchestrazione centralizzato, ogni nuovo banco introduce ulteriore variabilità e debito tecnico. Il rischio non è teorico: significa tempi lunghi per rispondere a non conformità, difficoltà a gestire modifiche e rollback delle sequenze, problemi in audit perché mancano prove robuste di cosa è stato fatto e con quali strumenti.

QX come piano di orchestrazione dei sistemi di test

QX nasce per trasformare l’insieme dei banchi prova in una **piattaforma** aziendale di Test & Measurement, in cui configurazioni, strumenti, sequenze e risultati sono gestiti in modo centralizzato e tracciabile. In questa architettura il framework locale (SPOT o PATH) si occupa di eseguire il test sul banco, mentre QX definisce chi può fare cosa, su quale postazione, con quali strumenti e con quali versioni di sequenza.

In scenari reali, difficilmente si parte da zero. Più spesso esiste un parco installato di sistemi LabVIEW e Python eterogenei, sviluppati nel tempo con approcci diversi e con livelli di strutturazione molto variabili. Il lavoro che facciamo come system integrator non è semplicemente “aggiungere QX sopra”, ma portare progressivamente questi sistemi dentro un modello coerente: normalizzare i dati di test, identificare e standardizzare le configurazioni, agganciare i flussi di esecuzione ai framework SPOT/PATH e collegarli al modello informativo di QX.

Questa transizione non è banale, ma è ciò che permette di passare da una costellazione di applicazioni legacy a un’infrastruttura in cui è possibile sapere in ogni momento quali prove sono disponibili su ogni banco, quali strumenti sono associati a quali postazioni, quali versioni di sequenze e parametri sono attivi e dove sono i dati.

Progettazione LabVIEW e Python orientata all’orchestrazione

Molti sistemi di test LabVIEW in esercizio sono stati progettati in un’epoca in cui l’obiettivo principale era “far funzionare il banco” e soddisfare i requisiti immediati di misura. Non erano pensati per essere orchestrati, né per esporre in modo pulito configurazioni, meta-dati e risultati a un livello superiore di governo.

Oggi, quando progettiamo o evolviamo applicazioni di test in LabVIEW o Python, il punto di partenza è diverso: il banco non è più il fine, ma il nodo di una rete. Questo significa progettare codice e architettura in modo

che si integrino nativamente con il framework di test (SPOT o PATH) e con QX, evitando soluzioni locali che funzionano solo finché qualcuno le conosce a memoria.

Nei progetti più complessi partiamo spesso da sistemi esistenti: analizziamo come sono stati strutturati, identifichiamo dove è possibile innestare SPOT o PATH come runtime di test, e come collegare le informazioni chiave a QX senza fermare la produzione. L'obiettivo non è riscrivere tutto, ma ridurre il debito tecnico rendendo governabili nel tempo le applicazioni di test e i dati che producono.

Dall'integrazione OT-IT alla qualità dei dati (e oltre)

La convergenza tra mondo OT e mondo IT è un presupposto, non un fine. Protocolli come OPC UA, API REST e integrazioni con MES, sistemi di qualità e applicazioni cloud permettono ai sistemi di test di scambiare dati con il resto dell'ecosistema aziendale in modo strutturato. Ma se ciò che esce dai banchi non è coerente, tracciabile e contestualizzato, l'integrazione rischia di propagare inconsistenze invece di portare valore.

QX agisce proprio su questo livello: non solo raccoglie risultati di test, ma li collega a configurazioni, strumenti, stato di taratura, versioni di sequenza, parametri di processo e contesto operativo. È questo che rende i dati realmente utilizzabili per analisi affidabili, per decisioni sul processo e, in prospettiva, per applicare algoritmi di intelligenza artificiale in modo non puramente sperimentale.

Intelligenza artificiale: perché senza orchestrazione è solo un esperimento

Negli ultimi anni si parla molto di IA applicata ai test e ai collaudi. Nella pratica, quello che vediamo sul campo è che senza orchestrazione e qualità dei dati, molti progetti rimangono POC isolati: interessanti in laboratorio, difficili da portare in produzione.

Affinché modelli di IA possano dare valore in questo contesto, servono dataset che non contengano solo "misure", ma anche il contesto completo: quale prodotto, con quale configurazione di test, con quali strumenti e in quale stato, con quale versione di sequenza. È esattamente questo tipo di informazione che QX centralizza e rende accessibile, rendendo possibile, ad esempio, individuare drift nei risultati correlati a strumenti o versioni di test, suggerire ottimizzazioni delle sequenze sulla base di dati storici o supportare la manutenzione predittiva dei sistemi di test.

Senza un layer come QX, l'IA sui test resta confinata a singoli banchi o progetti pilota; con un'infrastruttura orchestrata, può diventare una leva trasversale su più impianti e programmi.

Verso la gestione del ciclo di vita di strumenti e laboratori

In questo articolo ci siamo concentrati su come passare dall’automazione locale all’orchestrazione centralizzata di sistemi di test e dati.

Ma l’infrastruttura di **Test & Measurement** non si esaurisce con i banchi: include strumenti, tarature, manutenzioni, asset di laboratorio e tutti i processi collegati.

Si chiude così la serie dei quattro articoli dedicati ai sistemi di test, con un percorso che ha messo a fuoco la trasformazione dalle soluzioni locali a un’architettura orchestrata di Test & Measurement.