



Rational. software

Un nuovo approccio alla verifica e alla validazione dello sviluppo di dispositivi medici.

*Automatizzare i processi di sviluppo prodotti e conformità grazie al
software IBM Rational*

*Irv Badr, senior manager, Rational software,
IBM Software Group*

Indice
2 Introduzione
2 Affrontare le sfide peculiari del settore dei dispositivi medici
3 Approcci tradizionali alla conformità dei dispositivi
3 Verifica della progettazione del sistema
4 Validazione della progettazione del sistema
5 Verifica e validazione basate su modelli
8 L'approccio di verifica e validazione automatizzato in azione
9 Tool di sviluppo e gestione per processi automatizzati
10 Un processo di sviluppo migliorato a costo inferiore

Introduzione

L'elenco di funzionalità nei dispositivi medici è in rapido aumento: anche il più semplice dei dispositivi medici, come i sistemi di diagnostica e monitoraggio, ospita sempre più componenti di sistema, che aggiungono maggiore funzionalità al dispositivo a un costo contenuto. Ma, con l'aggiunta di componenti, il software del dispositivo diventa complesso e grava sulle attività di test della conformità e certificazione pre-commercializzazione, come la presentazione dei documenti (510(k)) prevista dalla Food and Drug Administration (FDA) statunitense.

Tuttavia, è possibile affrontare questi problemi utilizzando, nello sviluppo dei dispositivi medici, un processo di verifica e validazione basato su modelli. In primo luogo, questo processo può aiutare a gestire meglio la complessità del software, astrandolo come modello. In secondo luogo, il processo può aiutare a verificare e validare un sistema in evoluzione, sfruttando un modello eseguibile nelle primissime fasi del processo di sviluppo. Infine, la precoce validazione e verifica di un sistema può aiutare a ridurre il tempo di sviluppo totale e ad abbreviare il processo di presentazione alla FDA, un passo obbligatorio per immettere un dispositivo sul mercato statunitense.

Affrontare le sfide peculiari del settore dei dispositivi medici

Il flusso di lavoro tipico dello sviluppo di prodotti software embedded è spesso la raccolta dei requisiti, l'analisi, la progettazione del sistema, la progettazione dettagliata, il test e la gestione del progetto. Ma nel settore dei dispositivi medici, esiste una fase supplementare: la conformità. La FDA disciplina i prodotti sviluppati per il mercato statunitensi attraverso Quality System Regulations (QSR) 21 Code of Federal Regulations (CFR) Parte 820 §30, che prevede essenzialmente:¹

- *Conservazione della corretta documentazione in un file della cronologia di progettazione (Design History File, DHF).*
- *Conformità a Titolo 21 CFR Parte 11 che regola l'uso delle firme elettroniche nei DHF.*
- *Obbligo per i mercati internazionali di conformarsi all'International Organization for Standardization (ISO) 13485:2003 e di rispettare le disposizioni della Direttiva dell'Unione Europea concernente i dispositivi medici (DDM) 93/42.*
- *Conformità alle Buone Pratiche di Fabbricazione attuali (CGMP) e alle Buone Pratiche di Documentazione (GDP), entrambe dettate dalla FDA.*

Punti salienti

Tradizionalmente, i team di sviluppo usavano il codice sorgente per stabilire la tracciabilità tra i requisiti e la relativa implementazione.

Approcci tradizionali alla conformità dei dispositivi

Per soddisfare gli obiettivi di conformità, i team di sviluppo dei dispositivi medici creano un DHF all'inizio di un progetto. Il file può contenere requisiti informali, scritti a mano, e note di progettazione, nonché stampati ed estratti di codice sorgente di documenti formali di architettura e progettazione. Alcuni team sviluppo centrati sul codice sorgente comprendono anche documenti dei requisiti, documenti di progettazione formali ad hoc ed elenchi del codice sorgente, che rispecchiano il modo in cui i team implementano i requisiti come codice sorgente. Questo flusso di lavoro aiuta i team a stabilire la tracciabilità tra i requisiti e la loro esatta implementazione, come richiesto dai QSR. La tracciabilità può dimostrare che il dispositivo viene usato per lo scopo a cui è destinato; tuttavia, il metodo per stabilire la tracciabilità non è prescritto per legge. Poiché il codice sorgente è immediatamente disponibile, molti team lo utilizzano per la tracciabilità e come mezzo per rappresentare l'architettura e la progettazione del sistema. Questo metodo è comune tra i team che non dispongono di un approccio di modellazione formale nel processo di sviluppo.²

Verifica della progettazione del sistema

La Figura 1 mostra un processo di sviluppo raccomandato dalla Design Control Guide della FDA.³ Va notato che la conformità ai QSR può essere realizzata congiuntamente alla conformità alle GDP, aderendo alle fasi di sviluppo iterative, o a cascata, illustrate. I responsabili dei test possono eseguire la verifica del sistema rispetto ai requisiti, misurando l'output di progettazione rispetto all'input di progettazione.

L'input di progettazione comprende le specifiche definite dai requisiti dell'utente, che identificano l'uso previsto del dispositivo. Di solito comprende documenti di testo formali o semiformali e alcuni modelli che riflettono una serie di specifiche, rispetto alle quali il sistema deve essere costruito. L'output di progettazione, a sua volta, comprende le procedure definite dal produttore del dispositivo, che aiutano ad assicurare che il prototipo completato sia in linea con l'input di progettazione. Può comprendere, nel caso del software embedded, un elenco del codice sorgente che appartiene all'applicazione.

Punti salienti

Poiché i team di sviluppo devono assicurarsi di implementare le specifiche e di soddisfare gli obiettivi di progettazione, devono verificare il dispositivo misurando l'input di progettazione rispetto al relativo output.

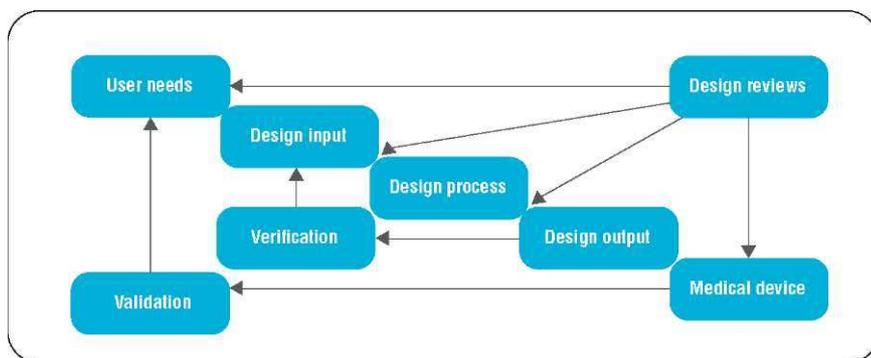


Figura 1: Una fase di verifica della progettazione controlla l'output di progettazione rispetto all'input.

Le milestone di input e output di progettazione costituiscono parte integrante del processo di sviluppo dei dispositivi medici. Poiché le milestone sono valide per un gran numero di dispositivi, nessun organismo di regolamentazione specifica o impone quali metodi di verifica della progettazione utilizzare. Di conseguenza, i produttori di dispositivi usano una serie di tool e processi a questo scopo e si affidano a documenti e ad elenchi del codice sorgente, come descritti per l'input e l'output di progettazione.

Validazione della progettazione del sistema

Dato che molti team considerano il codice sorgente come la misura definitiva dello stato di implementazione di un sistema, spesso esaminano a fondo il codice sorgente per predire il comportamento finale del dispositivo effettivo. In questo approccio centrato sul codice sorgente, i team eseguono l'applicazione sul dispositivo effettivo e analizzano le sue linee di codice sorgente per osservare il comportamento risultante e validare il sistema.

Punti salienti

Poiché verificare i dispositivi misurando l'input rispetto all'output può essere oneroso in termini di tempo e soggetto a errori, un approccio migliore alla verifica del sistema è modellare e raccogliere i dati del sistema per dimostrare l'uso previsto del dispositivo.

Seppur efficace, questo metodo di validazione del sistema si è rivelato costoso e soggetto a errori. I test unitari del sistema in evoluzione sul dispositivo di destinazione possono essere scomodi e lenti. Inoltre, potrebbe non essere possibile far girare il dispositivo attraverso tutti gli scenari d'uso previsti, senza sostenere pesanti spese o incontrare ostacoli logistici. Ad esempio, un hardware incompleto o inesistente potrebbe impedire il collaudo finale del sistema sul dispositivo. Inoltre, una piattaforma incompleta potrebbe restituire risultati difettosi per l'applicazione testata.

Man mano che i bug del software e del sistema vengono rilevati e corretti durante i test unitari, i team possono avere la necessità di aggiornare la progettazione e i requisiti associati per rispecchiare il codice sorgente modificato. A seconda delle dimensioni dell'applicazione, gli aggiornamenti possono essere onerosi in termini di tempo e soggetti a errori. E alcune modifiche potrebbero essere saltate, causando un errato sfasamento tra il DHF e il sistema implementato.

Verifica e validazione basate su modelli

Né i controlli di progettazione interni né i QSR richiedono di far funzionare effettivamente un dispositivo medico per verificarne e validarne il sistema. Per quanto riguarda la FDA, i dati del sistema raccolti possono dimostrare l'uso a cui il dispositivo è destinato. Questi dati possono essere raccolti dal dispositivo effettivo, nonché da un'esecuzione simulata del dispositivo – in effetti, i tool di sviluppo delle applicazioni automatizzati offrono incrementi di efficienza pressoché ineguagliati.

Punti salienti

In un approccio basato su modelli alla verifica e validazione della progettazione, è possibile stabilire la tracciabilità tra i componenti della progettazione e i requisiti e testare la progettazione in scenari difficili da creare.

La Figura 2 mostra un processo basato su validazione e verifica, imperniato sullo sviluppo iterativo di un output di progettazione basato su input di progettazione. La figura illustra inoltre una tracciabilità verificabile tra i requisiti e la validazione del sistema, e tra l'architettura e le attività di progettazione e verifica del sistema. I tool di modellazione possono fornire la tracciabilità dei requisiti e funzionalità di modello eseguibili automaticamente.⁴ Possono inoltre riferire la progettazione più recente per tutto il ciclo di vita del prodotto.

Inoltre, è possibile sfruttare un tool di gestione dei requisiti automatizzata per stabilire la tracciabilità tra i componenti della progettazione e i requisiti.

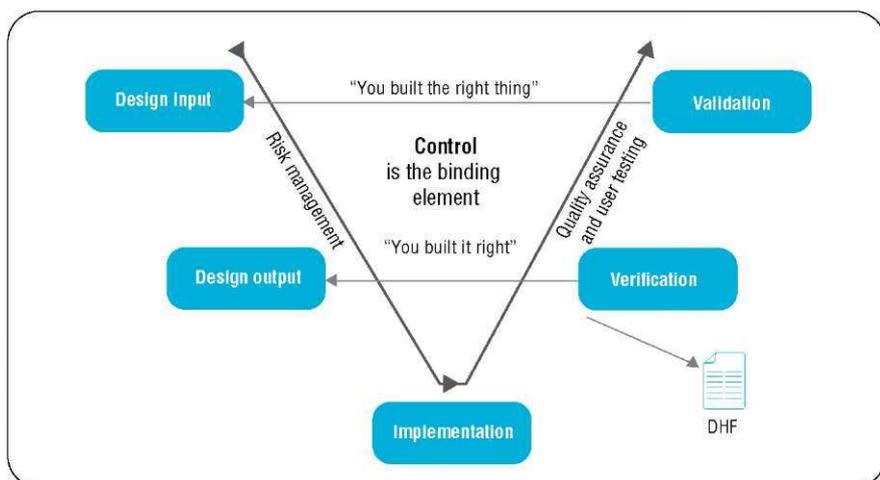


Figura 2: In un approccio tradizionale, il processo basato su validazione e verifica è imperniato sullo sviluppo iterativo di un output di progettazione basato su input di progettazione.

In un processo basato su modelli, è possibile architettare e progettare un'applicazione nel tool di modellazione.⁵ È possibile inoltre generare automaticamente casi di test unitari dai modelli di progettazione, come illustrato nella figura 3. Qui, i modelli sia per la progettazione che per il caso di test sono eseguibili.

Punti salienti

Il debugging del sistema tradizionale è molto più lento che in un approccio basato su modelli, perché il primo richiede molte più linee di codice – rispetto agli elementi di modelli – per implementare lo stesso sistema.

Innanzitutto, si deve verificare la progettazione sottostante, testare la progettazione e raffigurare gli scenari d'uso effettivi per il dispositivo. La maggior parte dei difetti e delle sviste di progettazione viene catturata durante questa fase di verifica del modello. I test unitari possono sottoporre la progettazione a possibili scenari, molti dei quali possono essere difficili da creare sul dispositivo reale.

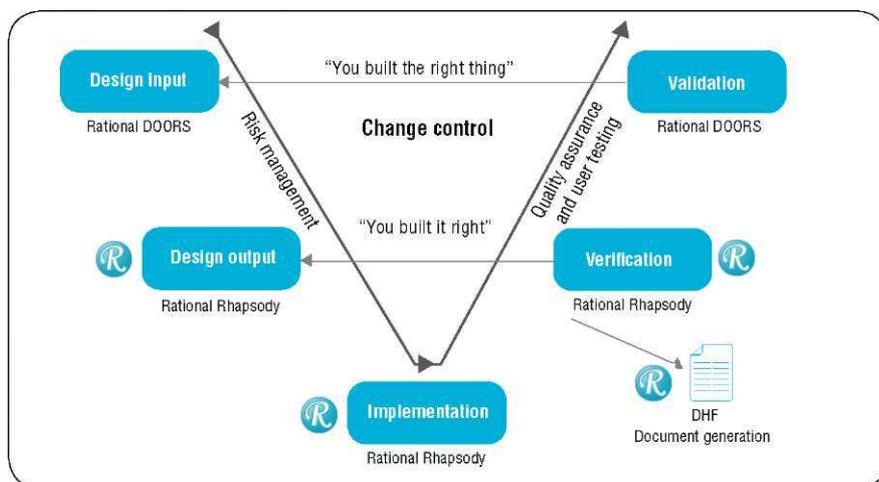


Figura 3: Tool di gestione dei requisiti automatizzata possono fornire automaticamente la tracciabilità dei requisiti e funzionalità di modello eseguibili e stabilire la tracciabilità tra componenti di progettazione e requisiti.

Poi, se si rilevano difetti o sviste, è possibile correggerli immediatamente nella progettazione, eliminando quindi la necessità di modificare manualmente il codice e la documentazione di progettazione. Nello sviluppo tradizionale, i difetti vengono corretti all'interno del codice sorgente, dopo averlo eseguito sul dispositivo reale. Ciò avviene successivamente nel processo di sviluppo. Di conseguenza, il debugging del sistema tradizionale è molto più lento che in un approccio basato su modelli, perché il primo richiede molte più linee di codice – rispetto agli elementi di modelli – per implementare lo stesso sistema.

Infine, per la validazione del sistema, è possibile tracciare automaticamente i casi di test rispetto ai requisiti operativi del sistema. Le funzionalità di tracciabilità dei requisiti a un tool di gestione dei requisiti formale e l'ambiente di modellazione automatizzano la tracciabilità. In altre parole, è possibile stabilire un processo di validazione e verifica pienamente automatizzato.

Punti salienti

Lo sviluppo basato su modelli è particolarmente importante per i dispositivi medici, perché consente ai responsabili dei test di garantire la sicurezza dei pazienti, durante eventi imprevisti o complessi, difficili da creare.

L'approccio di verifica e validazione automatizzato in azione

La Figura 4 è un progetto per un pulsossimetro, che utilizza un sensore a pinza applicato al dito per misurare i livelli di ossigeno nel sangue nonché la frequenza del polso. La progettazione mostra una macchina a stati, responsabile della de-pacchettizzazione dei dati del sensore. Immettendo dati del sensore reali o simulati, la macchina a stati può verificare rapidamente il funzionamento corretto.

La Figura 4 verifica inoltre che sia i livelli di ossigeno che la frequenza del polso rientrino negli intervalli di sicurezza. Questo caso di test è eseguito parallelamente alle verifiche di progettazione, per aiutare a garantire la sicurezza dei pazienti durante eventi imprevisti o complessi, difficili da creare. Ad esempio, i test unitari possono modellare il comportamento del pulsossimetro nell'improbabilità che non rilevi il polso durante il monitoraggio del livello di ossigeno del paziente. Per testare il pulsossimetro nello stesso scenario del mondo reale, occorrerebbe arrestare effettivamente il cuore del paziente.

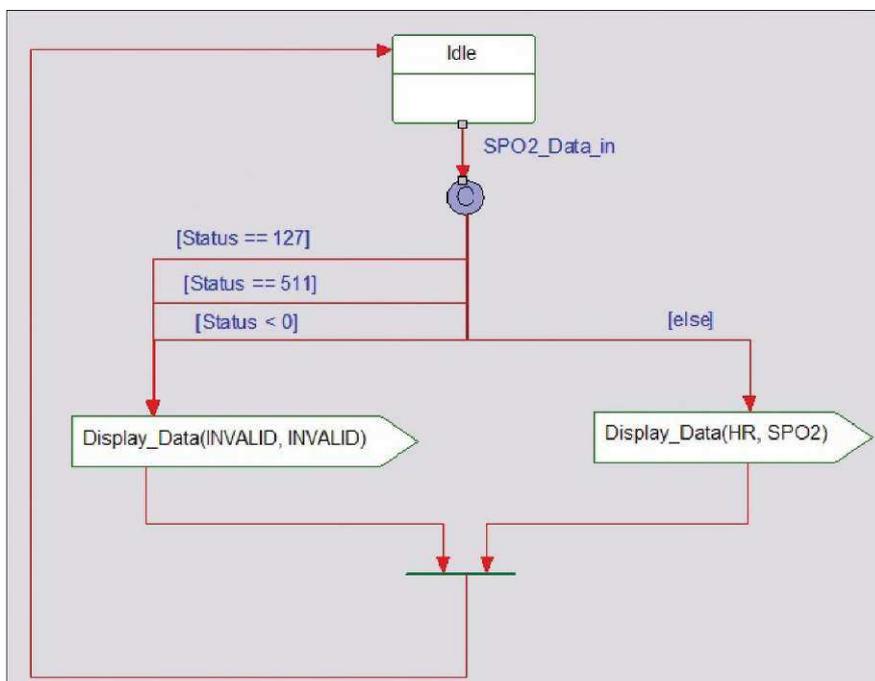


Figura 4: Utilizzando lo sviluppo dei dispositivi basato su modelli, è possibile modellare il comportamento di un pulsossimetro, nel caso in cui non rilevi il polso durante il monitoraggio del livello di ossigeno nel sangue. Per testare il pulsossimetro nello stesso scenario del mondo reale, occorrerebbe arrestare effettivamente il cuore del paziente.

Punti salienti

Il software per lo sviluppo del ciclo di vita Rational automatizza i processi e soddisfa i requisiti di validazione e verifica degli organismi di regolamentazione.

Tool di sviluppo e gestione per processi automatizzati

Il software IBM Rational® fornisce tool di sviluppo per il ciclo di vita, che affrontano i requisiti di validazione e verifica dei QSR. La piattaforma di gestione dei requisiti IBM Rational DOORS® può creare l'input di progettazione e gestire i requisiti del sistema e i test di validazione come serie di oggetti strutturati e tracciabili. Inoltre, Rational DOORS supporta le firme elettroniche conformi a 21 CFR Parte 11, consentendo quindi di usare legalmente e in modo più sicuro i record elettronici nei processi di progettazione e sviluppo.

Il software IBM Rational Rhapsody® è in grado di modellare il dispositivo medico per tutto il processo di progettazione – dalla definizione delle specifiche di sviluppo del sistema al rilascio del software sul dispositivo. Rational Rhapsody può essere utilizzato per modellare dispositivi medici complessi, per una qualunque combinazione di sistemi embedded e desktop. Il sistema embedded può essere un semplice pulsiossimetro, oppure un grande dispositivo, come un tomografo computerizzato (TC). Altri sistemi possono avere una serie di piattaforme interrelate, comprendenti computer headless desktop e workstation di monitoraggio con sistemi operativi convenzionali, quali piattaforme Linux®, UNIX® o Microsoft® Windows®.

Inoltre, Rational Rhapsody supporta la modellazione indipendente dal linguaggio e indipendente dal sistema operativo e può rilasciare gli stessi modelli praticamente su qualsiasi piattaforma. Questa è descritta come modellazione indipendente dalla piattaforma (Platform-Independent Modelling, PIM), dove un'unica serie di modelli può essere utilizzata su molte piattaforme diverse o indefinite. La PIM aumenta la produttività, consentendo di sfruttare i componenti progettati nelle generazioni future di piattaforme non ancora conosciute. Inoltre, la PIM consente di sviluppare la progettazione prima che sia disponibile l'hardware, permettendo di validare il comportamento funzionale tempestivamente nella progettazione, quando è meno oneroso correggere i difetti. È possibile poi validare le caratteristiche di progettazione specifiche per la destinazione, quando l'hardware diventa disponibile.

Punti salienti

Utilizzando i tool di verifica e validazione automatizzata Rational, è possibile ridurre i costi di sviluppo creando al contempo un dispositivo medico più affidabile, con meno rischi di guasto nell'impiego pratico.

Per i dispositivi medici indipendenti tradizionali, che utilizzano tipicamente un single-board computer embedded e un sistema operativo real-time, Rational Rhapsody è in grado di fornire supporto di architettura e progettazione oltre alla generazione automatica del codice sorgente. Queste funzionalità consentono un rapido re-targeting del modello all'esecuzione sull'host al dispositivo di destinazione. Il tool è in grado di supportare sistemi operativi real-time pronti per l'uso, o dispositivi privi di sistema operativo. Fornisce inoltre il debugging a livello di progettazione ospitato sul target, che può rivelarsi prezioso durante il processo di validazione e verifica, quando è necessaria la verifica a livello di target.

Rational Rhapsody può automatizzare la documentazione di progettazione estraendo le informazioni dal modello, attraverso template personalizzabili. Può poi documentare accuratamente l'implementazione della progettazione e fornire un iter cartaceo integrato, che parte dalle attività di gestione dei requisiti e modellazione per arrivare alla validazione e verifica.

Un processo di sviluppo migliorato a costo inferiore

Nella progettazione di dispositivi medici si possono soddisfare le linee guida e i regolamenti di progettazione QSR, oltre alle best practices di sviluppo del software e del sistema. Utilizzando un approccio di verifica e validazione automatizzato, è possibile ridurre i costi di sviluppo creando al contempo un dispositivo medico più affidabile, con meno rischi di guasto nell'impiego pratico. Inoltre, l'approccio automatizzato fornisce contenuto "live" per il DHF, che può essere automaticamente prodotto e gestito con i tool di sviluppo giusti.

La suite di soluzioni Rational per la gestione del ciclo di vita è ideata per automatizzare il processo di sviluppo attraverso la gestione dei requisiti; la modellazione del sistema e del software; e tool di test automatizzati, basati su modelli, come Rational Rhapsody e Rational DOORS.

Per ulteriori informazioni

Per saperne di più sul software IBM Rational Rhapsody e Rational DOORS, contattate il vostro rappresentante o Business Partner IBM, oppure visitate:

ibm.com/software/rhapsody

Nota sull'autore

Irv Badr ha quasi 20 anni di esperienza nello sviluppo di software embedded e tecnologia di modellazione. Ha progettato un'infrastruttura di comunicazione per dispositivi medici, nodi di networking, trasmissione digitale via cavo e controlli industriali. Irv è stato anche responsabile tecnico per la vendita e la commercializzazione di sistemi operativi real-time e tool di modellazione.

Dopo aver conseguito la laurea in ingegneria biomedica presso l'Università dell'Illinois, Irv ha conseguito un Master in Business Administration presso la Northwestern University. Attualmente, è senior manager per Rational software, un'unità di IBM Software Group.



Note

- 1, 4 Irv Badr, "Developing Platform Independent Embedded Applications," *Embedded Systems Magazine*, July 2005.
- 2, 5 Irv Badr, *Rapid Development through Agile Modeling*, Telelogic, February 2005.
- 3 U.S. Food and Drug Administration, "Design Control Guidance For Medical Device Manufacturers," March 11, 1997.

© Copyright IBM Corporation 2009

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Prodotto negli Stati Uniti d'America
Giugno 2009
Tutti i diritti riservati

IBM, il logo IBM, ibm.com, Rational, DOORS e Rhapsody sono marchi o marchi registrati di International Business Machines Corporation negli Stati Uniti, in altri paesi o in entrambi. Se questi e altri termini con il marchio IBM sono contrassegnati dal simbolo del marchio (® o ™), la prima volta che compaiono in queste informazioni, tali simboli indicano un marchio registrato o di fatto (*common law trademark*) statunitense di proprietà di IBM alla data della pubblicazione di queste informazioni. Tali marchi possono essere anche marchi registrati o marchi di fatto in altri paesi. Un elenco aggiornato dei marchi IBM è consultabile sul web in "Copyright and trademark information", sul [sito ibm.com/legal/copytrade.shtml](http://sito.ibm.com/legal/copytrade.shtml).

Microsoft e Windows sono marchi di Microsoft Corporation negli Stati Uniti, in altri paesi o in entrambi.

UNIX è un marchio registrato di The Open Group negli Stati Uniti e in altri paesi.

Linux è un marchio registrato di Linus Torvalds negli Stati Uniti, in altri paesi o in entrambi.

Gli altri nomi di aziende, prodotti o servizi potrebbero essere marchi di fabbrica o marchi di servizio di altri.

I riferimenti contenuti in questa pubblicazione a prodotti o servizi IBM non implicano che IBM intenda renderli disponibili in tutti i paesi in cui essa opera.

I dati contenuti nella presente documentazione sono forniti a scopo unicamente informativo. Anche se è stato compiuto ogni sforzo per verificare la completezza e l'accuratezza delle informazioni contenute in questa documentazione, le informazioni sono fornite così come sono ("as is"), senza garanzie esplicite o implicite di alcun tipo. Inoltre, queste informazioni si basano sui piani e sulla strategia dei prodotti attuali di IBM, che sono soggetti a modifica da parte di IBM senza preavviso. IBM non sarà responsabile di danni derivanti dall'uso, o altrimenti collegati a questa documentazione o a qualsiasi altra documentazione. Nulla di quanto contenuto in questa documentazione è inteso a, né dovrà avere l'effetto di creare garanzie o dichiarazioni da parte di IBM (o dei suoi fornitori o licenzianti), né di modificare i termini e le condizioni dei contratti di licenza applicabili, che regolano l'uso del software IBM.

I clienti IBM hanno la responsabilità di garantire la propria conformità ai requisiti di legge. È responsabilità esclusiva del cliente rivolgersi a un consulente legale competente riguardo all'identificazione e all'interpretazione di leggi e requisiti normativi pertinenti, che potrebbero influire sull'attività del cliente e su eventuali azioni che il cliente potrebbe dover intraprendere per conformarsi a tali leggi.

RAW14123-USEN-01