



# AGILE INTEGRATION:

UN MODELLO PER L'ARCHITETTURA AZIENDALE

**E-BOOK**

di Steve Willmott e David Codelli  
A cura di Deon Ballard

## **SOMMARIO**

<b>LA FINE DELLA PIANIFICAZIONE: ORGANIZZAZIONI E AGILITÀ .....</b>	<b>4</b>
<b>L'INFRASTRUTTURA DELL'AGILITÀ.....</b>	<b>6</b>
Integrazione distribuita.....	7
Container.....	9
Interfacce di programmazione delle applicazioni (API) .....	10
<b>L'ARCHITETTURA DELL'AGILE INTEGRATION .....</b>	<b>12</b>
Un lavoro di squadra .....	12
Architettura dell'infrastruttura.....	12
<b>PER UNA CULTURA DELL'ORGANIZZAZIONE AGILE.....</b>	<b>14</b>
<b>CONCLUSIONI: RAGGIUNGERE L'AGILE INTEGRATION .....</b>	<b>18</b>

**LA FINE DELLA PIANIFICAZIONE:  
ORGANIZZAZIONI E AGILITÀ**

**L'INFRASTRUTTURA DELL'AGILITÀ**

Integrazione distribuita

Container

Interfacce di programmazione delle applicazioni (API)

**L'ARCHITETTURA DELL'AGILE  
INTEGRATION**

Un lavoro di squadra

Architettura dell'infrastruttura

**PER UNA CULTURA**

**DELL'ORGANIZZAZIONE AGILE**

**CONCLUSIONI: RAGGIUNGERE**

**L'AGILE INTEGRATION**

La capacità di reagire alle innovazioni è sempre più spesso la chiave del successo aziendale. Attori rivoluzionari entrano nei mercati e nuove tecnologie stravolgono le aspettative dei consumatori: questi due fattori costringono le aziende a modificare i propri piani con ritmi sempre più rapidi. Per reagire in modo efficiente alle innovazioni ed emergere vincenti nei mercati di appartenenza, le aziende devono adottare processi e architetture software avanzati.

I container, l'integrazione distribuita e le interfacce di programmazione delle applicazioni (API) sono tre capacità fondamentali che costituiscono il nuovo framework alla base delle architetture, detto agile integration, all'interno del quale promuovono l'agilità e alimentano nuovi processi aziendali capaci di generare vantaggi competitivi.

In alcuni settori, come quello dei viaggi e dell'ospitalità, la trasformazione è stata innescata da modalità innovative di fare impresa: oggi vengono erogati nuovi servizi con i quali i consumatori interagiscono in modo differente. Questo trend verso cambiamenti rivoluzionari si propaga in altri settori strategici, dai servizi finanziari alla pubblica amministrazione, sostenuta da tecnologie e approcci moderni all'interazione tra aziende e clienti. Le nuove sfide impongono alle organizzazioni esistenti una trasformazione radicale delle tecnologie IT, per adeguarle all'erogazione dei nuovi servizi.

Per mantenere la posizione di mercato, le aziende devono poter pianificare e applicare in tempi rapidi le modifiche ai sistemi software.

Per adeguarsi ai ritmi accelerati di erogazione del software, l'azienda non può prescindere da un'infrastruttura agile. In questo ambito, "agile" non indica lo sviluppo rapido del software, ma assume un significato più tradizionale: flessibile, capace di muoversi con facilità.

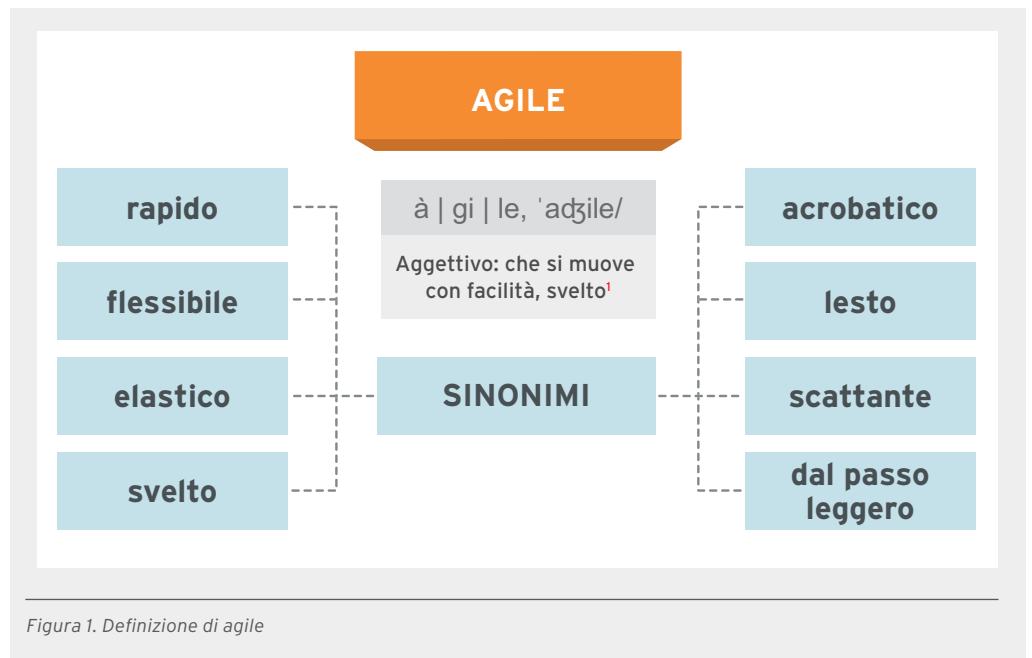


Figura 1. Definizione di agile

1 Dizionario Treccani

*“L'esigenza di agilità mirata a conquistare, conservare e offrire servizi ai propri clienti impone la presenza di interfacce sempre più agili, per connettere i sistemi di interazione e quelli di archiviazione.*

*Agili in termini di scalabilità, ma anche dotate della capacità di adattarsi rapidamente, ad esempio per consentire l'aggiunta di un nuovo attributo alle API esistenti e offrire più contesto in futuro”.*

**HENRY PEYRET**  
THE FORRESTER GROUP

*Peyret, Henry. “TechRadarTM: Integration Technologies, Q2 2015.” Forrester Research, Inc. 23 giugno 2015.*

Finora, le metodologie agili si sono concentrate sullo sviluppo software, puntando a migliorare e a ottimizzare la creazione delle applicazioni. Con l'approccio DevOps<sup>2</sup> si è cercato di integrare tale metodologia nel deployment delle applicazioni.

Di per sé, l'approccio DevOps in genere non va oltre questo, occupandosi principalmente delle nuove applicazioni software sviluppate all'interno dell'azienda.

Con un'infrastruttura agile è possibile guardare oltre e creare un ambiente che accoglie tutti i sistemi IT, incluso il software esistente. Quello dell'infrastruttura agile è un approccio che coglie la complessità dei sistemi esistenti, dei diversi tipi di dati, dei loro flussi e delle aspettative dei clienti e cerca un modo per unificare il tutto. Di fondo, quindi, si tratta di una questione di integrazione.

Un'organizzazione capace di mettere rapidamente a disposizione nuovi prodotti a prezzi migliori, è enormemente avvantaggiata rispetto a un'azienda che impiega tre mesi per l'implementazione e adotta una serie di fasi di verifica manuali.

Questa è l'agile integration: non una componente dell'infrastruttura, ma un approccio concettuale all'infrastruttura che include dati e applicazioni, hardware e piattaforme. Dall'allineamento di tecnologie di integrazione, DevOps e metodologie agili, ha origine una piattaforma che offre ai team la capacità di reagire ai cambiamenti assecondando i ritmi imposti dal mercato.

## **LA FINE DELLA PIANIFICAZIONE: ORGANIZZAZIONI E AGILITÀ**

“La pianificazione che noi conosciamo è un processo obsoleto”. È il messaggio del discorso di apertura lanciato da Jim Whitehurst, CEO di Red Hat, durante il Red Hat Summit del 2017. “In un ambiente costantemente mutevole, pianificare è una prassi ormai inefficiente”.<sup>3</sup> L'accelerazione e il cambiamento dei contesti aziendali scuotono ogni scenario e costringono a mutare rapidamente i piani. Rimanere vincolati a un piano d'azione può avere un costo elevato.

In questo senso i piani di un'azienda hanno meno valore quando le informazioni sono scarse o l'ambiente non è sufficientemente stabile.

### **La scarsa conoscenza è un limite**

Pianificare un'infrastruttura è un'attività a lungo termine, frequentemente estesa su più anni. I piani pluriennali, però, possono rivelarsi controproducenti e limitare la capacità di innovare o di adattarsi alle variazioni del mercato. La "fine" della pianificazione a cui allude Jim Whitehurst indica proprio l'esigenza di una pianificazione più rapida, ma che possa comunque essere rispettata. I piani hanno minori aspettative di vita, ma sussistono in un ambiente capace di generarne di nuovi.

Ovviamente, quando i team sono abituati a cicli di sviluppo compresi tra i 6 e i 24 mesi, adattarsi a questi cambiamenti rapidi è complicato. Il problema è ancora più sentito nelle organizzazioni con strutture tradizionali in competizione con le start-up, che affrontano il mercato in modi totalmente nuovi. Netflix, Blockbuster o Uber sono esempi lampanti, ma l'effetto rivoluzionario delle start-up risale ai primi tempi dell'Era dell'informazione, con Amazon nel 1993 o i personal computer degli anni Ottanta.

<sup>2</sup> Innovate faster with DevOps <https://www.redhat.com/it/insights/devops>

<sup>3</sup> Discorso di apertura di Jim Whitehurst al Red Hat Summit 2017.

<https://www.cbronline.com/news/enterprise-it/software/red-hat-ceo-planning-know-dead/>

TABELLA 1: PRINCIPALI RIVOLUZIONI PER SETTORI

SETTORE	SERVIZIO TRADIZIONALE	ELEMENTO RIVOLUZIONARIO	EFFETTI
Trasporti	Taxi, trasporto pubblico	Uber, Lyft	Creazione di un'esperienza cliente uniforme, impossibile da replicare per le piccole imprese locali
Gestione patrimoniale	Società di investimenti	Fondi automatici	Riallocazione dei differenziatori per la gestione dei fondi dal personale agli algoritmi
Vendita al dettaglio	Shopping fisico	Amazon	Cambiamento delle abitudini di acquisto, con il passaggio dall'acquisto in un esercizio fisico a quello online
Motori di ricerca	Google, ricerca tramite browser	Ricerca vocale	Ha effetti sul principale canale di Google verso il mercato e consente l'accesso di nuovi attori

Start-up e aziende rivoluzionarie hanno il vantaggio della piena libertà di strutturare l'infrastruttura, i team, le applicazioni, l'architettura e perfino i processi di deployment. Non si tratta solo di avere idee innovative: significa metterle in pratica, perché non sono vincolate a un'infrastruttura esistente, o, per dirla con le parole di Rachel Laycock, "ai team legacy".<sup>4</sup> Possono essere davvero agili.

Oltre a ideare qualcosa di nuovo, queste organizzazioni realizzano anche sistemi pronti al cambiamento. L'infrastruttura software è parte integrante della loro capacità di differenziarsi: quasi ogni componente del sistema può essere sostituito, aggiornato o rimosso, per adattarsi alle nuove realtà del mercato. Tuttavia, con il passare del tempo, alcune start-up risentono di una ridotta capacità di adattamento, mentre le migliori provvedono a far sì che la propria adattabilità sia protetta a tutti i costi.

#### Affrontare le sfide

Negli ambienti a rapida evoluzione, la chiave del successo è il funzionamento agile dell'intera infrastruttura IT.

Il cambiamento deve avvenire su due livelli:

- A livello organizzativo e culturale con il supporto dei processi agili, dal design dell'architettura alla comunicazione dei team.
- A livello tecnico, con un'infrastruttura che consente di aggiornare, aggiungere e rimuovere rapidamente le capacità.

Di per sé, i cambiamenti tecnici e culturali non creano l'agilità ma ne costituiscono le fondamenta.

Marty Cagan, Product Manager di eBay, applica una sorta di "tassa" ad ogni progetto: parte del tempo e delle risorse impiegate in ogni progetto ordinario vengono destinate verso nuovi progetti infrastrutturali.<sup>5</sup> Ciò rende prioritari i nuovi progetti e l'innovazione.

<sup>4</sup> Rachel Laycock, ["Continuous Delivery"] Sessione pomeridiana generale, Red Hat Summit - DevNation 2016. 1 luglio 2016, San Francisco, California. <https://youtube.com/watch?v=y87SUSOfqTY>

<sup>5</sup> Cagan, Marty, "Inspired: How to Create Products Customers Love." Wiley Press, 2017

*"Chi non riesce a superare i propri concorrenti in termini di time-to-market e agilità, ha perso in partenza. Le nuove funzionalità sono sempre una scommessa. Se si è fortunati, il 10% darà i risultati desiderati. Quindi, prima si riesce a immetterle sul mercato e a testarle, maggiore sarà il vantaggio competitivo. Anche l'azienda ottiene più rapidamente un ritorno sugli investimenti, iniziando a guadagnare prima."*

**GENE KIM**  
THE PHOENIX PROJECT

*Gene Kim, Kevin Behr e George Spafford. The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win. Portland, Oregon: IT Revolution Press, 2013.*

## L'INFRASTRUTTURA DELL'AGILITÀ

Alzare barriere alle nuove tecnologie non aiuta a creare un'infrastruttura agile, in quanto i diversi team si muovono in diverse direzioni alla ricerca di possibilità di miglioramento. Senza un insieme coerente di obiettivi a cui puntare, stabilire quali nuove capacità fanno davvero la differenza rispetto al funzionamento complessivo dell'azienda può diventare complicato.

### I tre pilastri dell'agile integration

L'approccio all'agile integration si fonda su tre tecnologie principali.

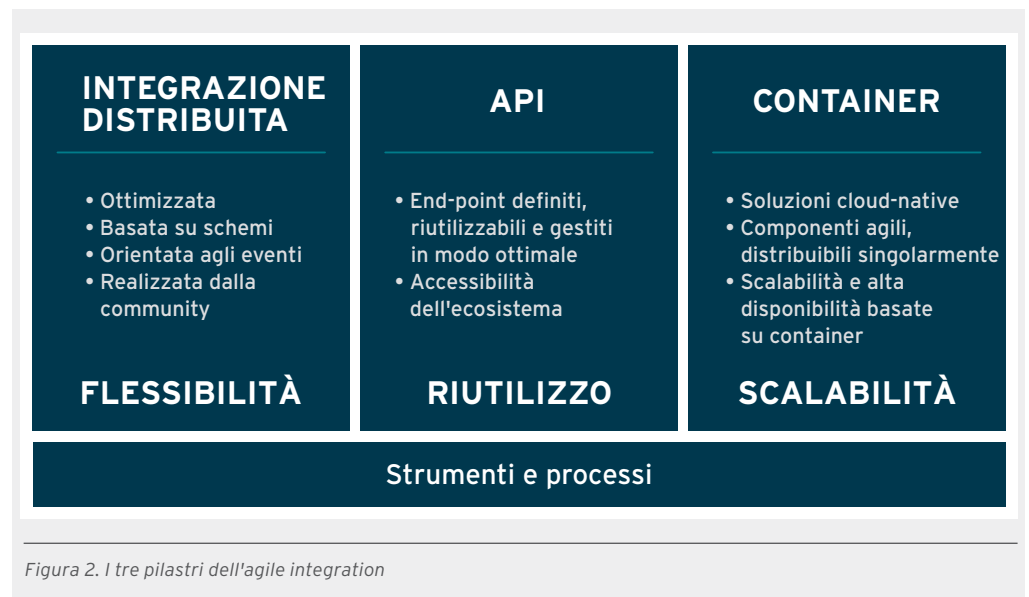


Figura 2. I tre pilastri dell'agile integration

- 1. Integrazione distribuita.** In genere le attività e i flussi di dati dell'azienda possono essere rappresentati da qualche decina di modelli di integrazione ad alto livello. Se inseriti nei container, questi modelli possono essere distribuiti secondo le dimensioni e negli ambienti richiesti da applicazioni e team specifici. È il concetto di architettura di integrazione distribuita, diversa dalla tradizionale architettura di integrazione centralizzata, che consente ai singoli team di definire e distribuire i modelli di integrazione necessari in modo agile.
- 2. API.** API stabili e ben gestite sono fondamentali per la collaborazione tra team e devops. Le API raggruppano le risorse chiave in interfacce stabili e riutilizzabili, che funzionano come moduli da riutilizzare nell'organizzazione oppure con i partner e i clienti. Il loro deployment può essere effettuato insieme ai container in ambienti diversi, consentendo agli utenti di interagire con diversi insiemi di API.
- 3. Container.** I container rappresentano la piattaforma di deployment di base per le API e le tecnologie di integrazione distribuita. Consentono di erogare uno specifico servizio in un ambiente specifico, semplificandone lo sviluppo, il test e la manutenzione. Piattaforma principale degli ambienti DevOps e dei microsistemi, i container usati per l'integrazione consentono una relazione trasparente e collaborativa tra i team di sviluppo e quelli di infrastruttura.

Queste tre capacità sono fondamentali per conferire agilità all'infrastruttura IT, poiché garantiscono un livello di astrazione elevato, al quale i diversi team possono collaborare. L'uso di una piattaforma per container con API e integrazioni distribuite consente di distinguere l'implementazione dell'integrazione dall'integrazione stessa. I team ottengono maggiore agilità perché sia le API sia i modelli di integrazione distribuita possono contenere risorse chiave specifiche a un livello ampiamente comprensibile, senza che sia necessario comprendere o modificare l'architettura alla base.

Presa singolarmente, ognuna di queste tecnologie può fornire un'agilità significativa per specifiche sfide di integrazione. Nel loro insieme, avranno un effetto amplificato. Mettere in risalto la tecnologia è un aspetto culturale: i vantaggi che essa offre si moltiplicano se combinati alle strategie DevOps, soprattutto alle procedure di automazione e deployment.

### **Integrazione distribuita**

Una delle principali sfide dei sistemi IT odierni è data dall'esigenza di connettere applicazioni utilizzate da diversi team. La difficoltà di integrare i servizi porta alla creazione di hub di integrazione sempre più centralizzati e complessi. Tali hub, spesso implementati come ESB (Enterprise Service Bus), diventano dei colli di bottiglia estremamente complessi e rigidi che ostacolano il cambiamento rapido.

Con l'integrazione distribuita si raggiungono molti degli obiettivi tecnici degli ESB di vecchia generazione, ma secondo modalità più adattabili ai vari team di un'organizzazione. Come con gli ESB, la tecnologia di integrazione distribuita è dotata di capacità di trasformazione, routing, analisi, gestione degli errori e invio di notifiche di avviso. La differenza sta nell'architettura dell'integrazione.

Nell'architettura di integrazione distribuita, ogni punto di integrazione è considerato come un deployment unico e distinto, e non come parte di un'applicazione di integrazione centralizzata e di grandi dimensioni. L'integrazione può essere containerizzata e distribuita in locale per uno specifico progetto o team, senza che ciò influisca sulle altre integrazioni distribuite in azienda. È un approccio che assicura la flessibilità richiesta dai progetti agili. Utilizza, inoltre, la stessa toolchain dei team agili o DevOps, sfruttando la stessa piattaforma di base dei container; in questo modo aumenta la capacità dei team di gestire le proprie integrazioni secondo strumenti e tempistiche proprie. Di fondo, considera l'integrazione come se fosse un microservizio<sup>6</sup> e ciò accelera la velocità di sviluppo e il rilascio delle integrazioni.

Strategico è anche l'allineamento con gli strumenti e i processi degli sviluppatori. Un aspetto chiave dell'integrazione distribuita è che non si tratta di un'infrastruttura software centralizzata, sviluppata e gestita da un gruppo di utenti specializzati e poi distribuita separatamente rispetto al processo di sviluppo del software. Quando l'architettura di integrazione viene distribuita mediante strumenti e piattaforme comuni, diventa accessibile a tutti gli sviluppatori a livello di progetto e agevola il deployment ottimizzato sempre e ovunque sia necessario.

---

<sup>6</sup> Definizione di microservizi (Martin Fowler) <https://martinfowler.com/articles/microservices.html>

*“Nel software, quando qualcosa non funziona a dovere, la soluzione per uscire dall'impasse non è ignorare il problema, ma ripeterlo più spesso”.*

**DAVID FARLEY**

CONTINUOUS DELIVERY: RELIABLE SOFTWARE RELEASES THROUGH BUILD, TEST, AND DEPLOYMENT AUTOMATION

*David Farley and Jez Humble, Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation. Addison-Wesley Professional, 2010.*

**TABELLA 2. CONFRONTO DELLE TECNOLOGIE DI INTEGRAZIONE PER OGNI FASE DEL CICLO DI VITA DEL SOFTWARE**

FASE DEL CICLO DI VITA	ESB, GRAN PARTE DELLE IPAAS (INTEGRATION PLATFORM-AS-A-SERVICE)	SUPPORTO PER TECNOLOGIE DI INTEGRAZIONE DISTRIBUITA
Controllo versione	Proprietaria	Github e altri
Creazione	Proprietaria	Maven e altri
Adozione	Proprietaria	Container e altri strumenti DevOps
Gestione e scalabilità	Proprietaria	Container e altri strumenti DevOps

L'impiego di un ESB impone al team l'utilizzo degli strumenti dell'ESB per il suo intero ciclo di vita, in aggiunta a quelli già utilizzati negli ambienti di sviluppo e operativi. È una limitazione che complica le attività e le rende inefficienti e soggette a errori.

### La messaggistica consolida l'integrazione

A livello di architettura, l'integrazione distribuita considera le integrazioni come microservizi. Possono infatti essere containerizzate, sono distribuibili in locale e in modo semplice, e possono avere cicli di rilascio rapidi.

La tecnologia di integrazione deve poter supportare questo tipo di architettura ottimizzata e basata su microservizi. Con Red Hat® Fuse gli utenti possono utilizzare le integrazioni come un codice eseguibile ovunque, anche in un container.

Inoltre, Fuse è abbinato a Red Hat JBoss AMQ e offre quindi un'infrastruttura per la messaggistica. Questa funzionalità, se consolidata, garantisce un efficace instradamento di dati ed eventi tra i sistemi. La messaggistica, come i microservizi, è un importante strumento dell'architettura, poiché la sua natura asincrona non richiede dipendenze.

La combinazione di integrazioni e messaggistica migliora le prestazioni complessive dell'architettura di integrazione, grazie ad un routing più efficiente, al supporto per più linguaggi e protocolli, al throughput asincrono e a una migliore gestione dei dati.

*“Una nuova rivalità tra concorrenti, a cui spesso ci si riferisce come trasformazione digitale, spinge le organizzazioni a ripensare la propria architettura IT, a ridistribuire i carichi di lavoro sulle infrastrutture, i cloud e gli elementi on-premise e ad agevolare l'interoperabilità a supporto delle operazioni e della strategia aziendale. Sono cambiamenti che esigono un nuovo approccio all'integrazione, che noi chiamiamo "integrazione ibrida".*

**CARL LEHMANN**  
451 GROUP

*Carl Lehman, 451 Research, "The Disruptive Role of Integration PaaS and APIs in the New Hybrid Integration Platform Market." Luglio 2017.  
<https://451research.com/report-long?icid=3862>.*

### Un trend da seguire

I dati segnalano che l'adozione dei container è in aumento, ma di quanto? E con quale finalità? Gli analisti di 451 Research stimano una crescita del mercato pari al 250%<sup>7</sup>, ma fanno riferimento alla spesa e non ai deployment, che di fatto sono più difficili da calcolare. Da un'indagine condotta da Bain e commissionata da Red Hat emerge che attualmente circa il 20% dei clienti distribuisce i container in produzione e più o meno la stessa percentuale in ambienti di sviluppo e test; oltre il 30% sta ancora valutando la tecnologia o è nella fase di proof of concept.<sup>8</sup>

Occorre comprendere bene cosa significa adottare i container. Enterprisers Project indica quattro diversi modelli di impiego dei container: come piattaforma generica di sviluppo o deployment, come piattaforma cloud-native o di microservizi, con un cloud ibrido o per progetti innovativi.<sup>9</sup> La modalità di impiego dei container influisce sulle ragioni alla base dell'adozione.

L'agile integration ha come obiettivo la realizzazione di una piattaforma infrastrutturale capace di supportare le operazioni esistenti, e benché prenda in prestito elementi dai diversi modelli di implementazione, si tratta di una piattaforma e, in quanto tale, serve da base per i nuovi progetti e per i servizi già in essere.

### Container

Virtualizzazione, cloud e container sono tecnologie simili che puntano a raggiungere obiettivi simili. Separano l'ambiente operativo software dall'hardware fisico, agevolando così la sovrapposizione di più istanze sull'hardware, rendendo più efficace utilizzo, scalabilità e distribuzione. Ovviamente, affrontano la sfida in modi diversi. La virtualizzazione astrae il livello del sistema operativo; il cloud elimina il concetto di istanze del server distinte e permanenti, mentre i container definiscono l'ambiente operativo e le librerie essenziali per l'esecuzione di una singola applicazione.

È l'approccio regolato e ottimizzato che rende la tecnologia dei container lo strumento ideale per gli ambienti software moderni. Ogni istanza utilizza una definizione invariabile, dal sistema operativo alla versione esatta di ogni libreria. Per ogni istanza, quindi, l'ambiente è altamente ripetibile e coerente, idoneo anche ai flussi di integrazione continua/distribuzione continua (CI/CD). Poiché l'immagine del container definisce solo gli elementi necessari a una singola applicazione, i container sono associati ai microservizi e la loro orchestrazione è funzionale anche all'orchestrazione della distribuzione e della gestione di infrastrutture di microservizi di grandi dimensioni.

Ottimizzazione e ripetibilità fanno dei container la piattaforma tecnologica ideale per l'agile integration.

<sup>7</sup> Infografica di 451 Research basata sul report "Cloud-Enabling Technologies Monitor", gennaio 2017.  
[https://451research.com/images/Marketing/press\\_releases/Application-container-market-will-reach-2-7bn-in-2020\\_final\\_graphic.pdf](https://451research.com/images/Marketing/press_releases/Application-container-market-will-reach-2-7bn-in-2020_final_graphic.pdf)

<sup>8</sup> Indagine Bain: For Traditional Enterprises, the Path to Digital and the Role of Containers, novembre 2016.  
<https://www.redhat.com/it/resources/path-digital-containers>

<sup>9</sup> <https://enterpriseproject.com/article/2017/8/4-container-adoption-patterns-what-you-need-know>

I tradizionali approcci all'integrazione presentano una struttura fortemente centralizzata, con ESB posizionati nei punti focali dell'infrastruttura. L'integrazione distribuita e la gestione delle API hanno entrambe un'architettura decentralizzata, che distribuisce solo la funzionalità richiesta ove necessario, o a un team specifico. I container possono essere la piattaforma alla base di entrambi gli approcci, grazie alla loro naturale invariabilità che assicura la coerenza delle immagini e dei deployment nei diversi ambienti: possono essere rapidamente distribuiti o sostituiti senza conflitti o dipendenze non trasparenti.

In un'architettura distribuita, che sia con integrazioni o con API, è fondamentale che la progettazione e l'implementazione di nuovi servizi non richiedano un processo di approvazione complesso.

I container consentono tanto alle API quanto alle integrazioni distribuite di essere utilizzate come microservizi. Forniscono strumenti comuni ai team di sviluppo e a quelli operativi, e consentono l'impiego di processi di sviluppo rapidi e procedure di rilascio gestite.

### **Container e orchestrazione**

Ogni container rappresenta un singolo servizio o applicazione, così come un microservizio rappresenta una singola funzionalità distinta dalle altre. In un'architettura di microservizi possono coesistere decine o centinaia di servizi separati, che vengono duplicati negli ambienti di sviluppo, test e produzione,

con un numero di istanze conseguentemente elevato. Affinché l'ambiente containerizzato sia efficace, la capacità di orchestrare e di eseguire attività di amministrazione avanzate è strategica.

Red Hat OpenShift combina i container Docker con il progetto di orchestrazione Kubernetes di Google, e prevede un'amministrazione centralizzata che include gestione delle istanze, monitoraggio, logging, gestione del traffico e automazione, attività non consentite da un ambiente di soli container.

Red Hat OpenShift include anche strumenti pensati per gli sviluppatori, come cataloghi self-service, clustering delle istanze, persistenza delle applicazioni e isolamento a livello di progetto,

in una combinazione che soddisfa tanto i requisiti operativi, in particolare stabilità e test, quanto le esigenze degli sviluppatori in termini di facilità d'uso e rapidità di erogazione.

### **Interfacce di programmazione delle applicazioni (API)**

La maggior parte delle infrastrutture informatiche contiene centinaia, se non migliaia, di sistemi, applicazioni e risorse, che interagiscono tra di loro con difficoltà. Spesso gli amministratori non riescono neanche a comprendere quali sistemi siano disponibili.

Le API sono le interfacce per tutte le risorse collegabili tramite tecnologie di integrazione. Costituiscono inoltre un sistema di definizioni o regole che stabiliscono come le applicazioni comunicano tra loro.

Mano a mano che le organizzazioni passano da un approccio all'integrazione centralizzato a uno distribuito, il self-service diventa una priorità. Autorità e autonomia sono indispensabili per i team agili che devono ricercare, testare e usare i servizi sviluppati all'interno e all'esterno della propria azienda. Si tratta di caratteristiche che possono essere fornite con solide funzionalità API, e che offrono ai team l'integrazione necessaria mentre l'organizzazione continua a garantire la gestione e l'applicazione di sicurezza, autorizzazioni e criteri d'utilizzo. Per i team, le API rappresentano un modello di riferimento per la progettazione delle integrazioni.

Sono diverse da un'applicazione definitiva in quanto determinano le modalità di interazione delle applicazioni e possono quindi essere usate dai singoli sviluppatori come moduli all'interno dei loro progetti. Le API offrono un linguaggio comune a sviluppatori e team e possono anche promuovere la creazione di community, che collaborano allo scopo di individuare utilizzi innovativi dei servizi.

È possibile esporre API o sottoinsiemi di API a destinatari diversi, perché le esigenze di un fornitore possono differire da quelle dei team di sviluppo interni o degli sviluppatori delle community. La gestione delle API prevede la progettazione dell'API per l'applicazione e il gruppo di utenti, nonché la gestione dell'intero ciclo di vita dell'interfaccia. Poiché le interfacce di programmazione delle applicazioni sono frequentemente considerate come prodotti, ognuno affidato a un team diverso, emerge la necessità di garantire uniformità e semplicità d'uso delle diverse risorse.

Come con l'integrazione distribuita, i container possono fungere da piattaforma per lo sviluppo, il deployment e la gestione delle API, secondo modalità che allineano lo sviluppo delle API a processi e strumenti di sviluppo e operativi di più ampia portata.

### La giusta piattaforma API può incrementare la produttività degli sviluppatori

Le API sono efficaci se possono essere utilizzate da altri sviluppatori interni e utenti esterni. Red Hat 3scale API Management Platform include strumenti utili per ogni tipo di utente: un portale per gli sviluppatori che desiderano collaborare alla creazione di API e uno per gli amministratori, nel quale le API vengono pubblicate.

La piattaforma aiuta a rendere disponibili le API all'esterno, grazie alle funzioni di autenticazione, all'integrazione con i principali provider di servizi cloud e alla possibilità di esecuzione all'interno dei container.

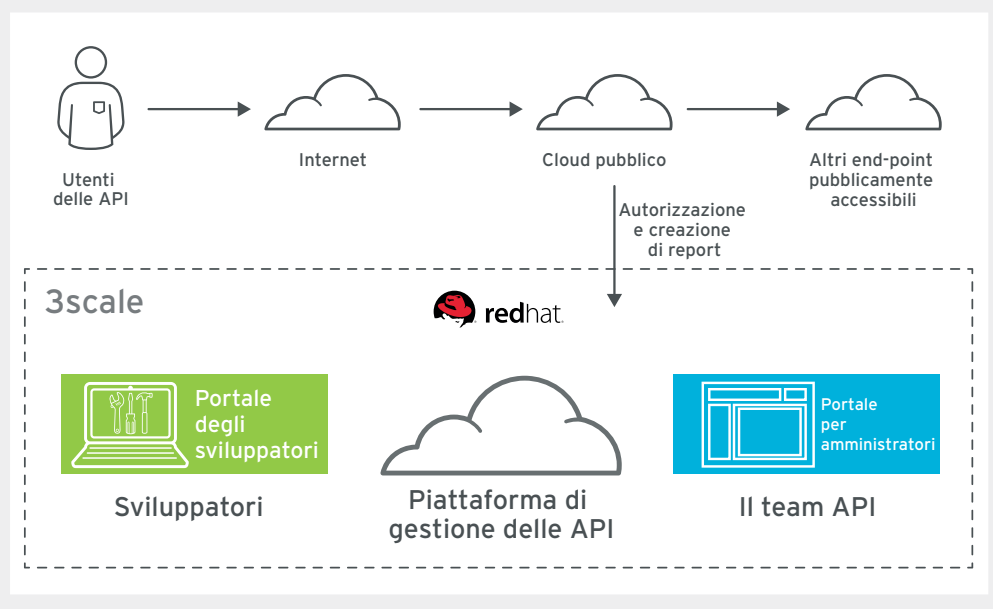


Figura 3. Panoramica sulla gestione delle API, gli end-point e il cloud pubblico

Una strategia ottimale per le API prevede la progettazione dell'API e le modalità di distribuzione. 3scale API Management Platform, soprattutto se implementata in una piattaforma per container, offre gli strumenti idonei per l'esecuzione di tale strategia.

## L'ARCHITETTURA DELL'AGILE INTEGRATION

### Un lavoro di squadra

Se distribuite e disponibili ai team come capacità riutilizzabili, le tecnologie su cui si basa l'agile integration sono più efficaci.

Con capacità si intende che gruppi di utenti autorizzati possono utilizzarle in modo autonomo, seguire con facilità le linee guida aziendali e accedere alle procedure ottimali. Spetta agli architetti o agli amministratori IT definire processi chiari per i singoli team, ad esempio:

- Fornire linee guida all'utilizzo ampiamente disponibili.
- Far applicare regole di utilizzo e procedure ottimali quando necessario, ma concedere la libertà di sperimentare.
- Definire procedure idonee per ogni fase del ciclo di vita: prototipo, test, entrata in produzione, aggiornamento e ritiro.
- Agevolare la condivisione delle informazioni per nuovi deployment e sviluppi.
- Consentire ai team dell'infrastruttura di abilitare e distribuire capacità self-service, anziché coinvolgerli in ogni processo.

A un team dedicato al software deve essere concesso, ad esempio, di sviluppare, testare e preparare una nuova API per il rilascio in modo completamente autonomo, stabilendo processi idonei ad aggiornare gli altri gruppi e a creare documentazione. Prima della pubblicazione o dell'entrata in produzione possono essere attuate procedure o controlli incrociati con altri team, ma l'infrastruttura deve prevedere la massima automazione del processo.

### Architettura dell'infrastruttura

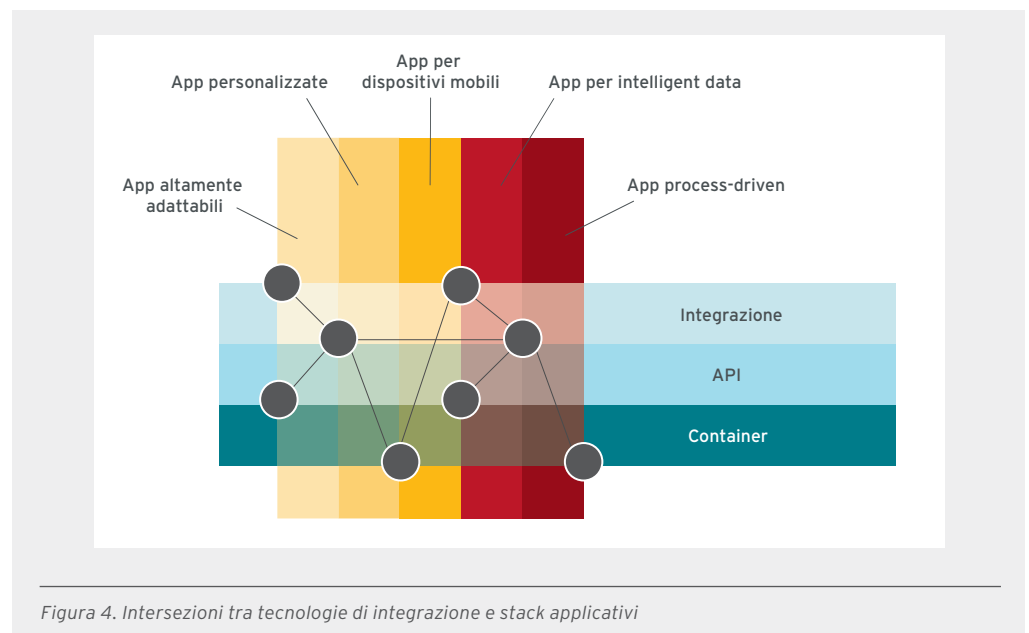


Figura 4. Intersezioni tra tecnologie di integrazione e stack applicativi

Nel complesso, container, API e integrazioni rappresentano le solide fondamenta degli ecosistemi software di un'azienda e, in molti casi, sono anche i punti di accesso per le integrazioni offerte da provider esterni.

Diversi tipi di sistema espongono diversi end-point riutilizzabili, ciascuno mostrato come API riutilizzabile e molti eseguiti nei container, per garantire scalabilità e facilità di deployment. Le integrazioni offrono capacità di trasformazione, composizione e logiche di business assimilate, ovunque all'interno del sistema, integrando un gruppo di singoli servizi o raccogliendo i risultati derivanti dai diversi ambiti aziendali.

È poi possibile aggregare ulteriormente le applicazioni integrate prima di destinarle alle applicazioni per l'utente finale.

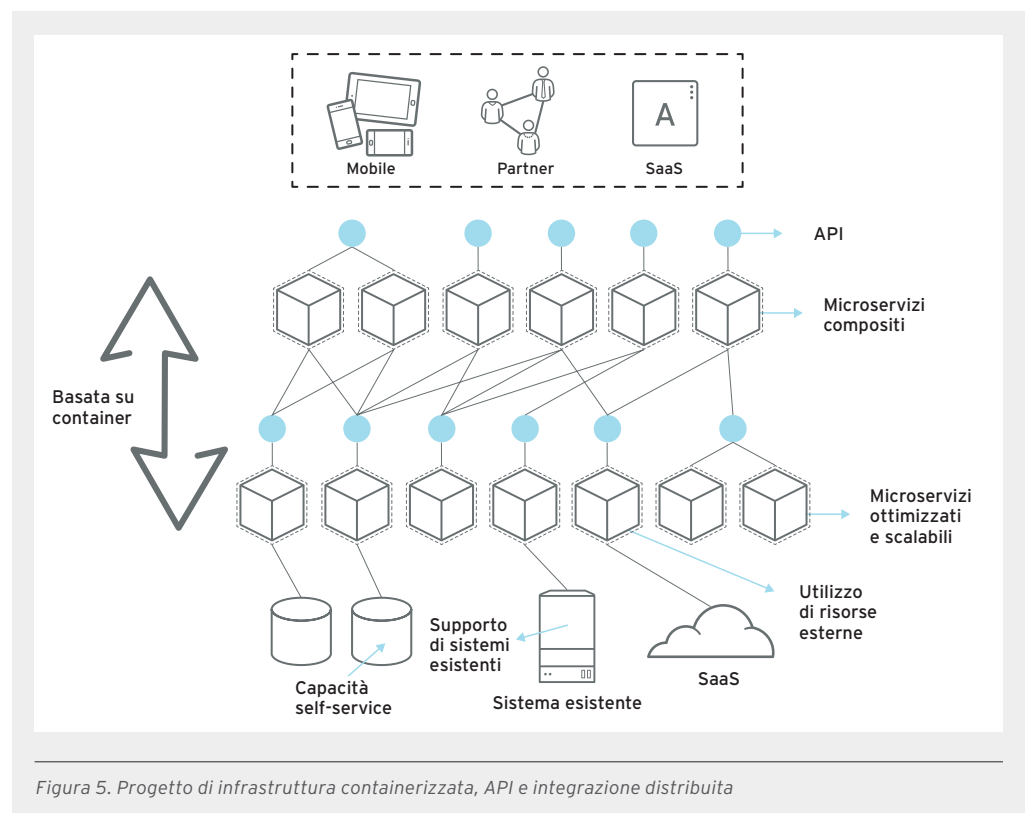


Figura 5. Progetto di infrastruttura containerizzata, API e integrazione distribuita

Non è necessario che tutti i sistemi vengano destrutturati in componenti sempre più piccoli o passino attraverso i molteplici livelli di astrazione delle API. Operazioni di questi tipo possono infatti ridurre l'efficienza, aggiungere latenza o complessità inutili. In alcuni ambiti, conservare la funzionalità ESB esistente può essere la scelta giusta per mantenere le connessioni tra specifiche applicazioni. Occorre anche tener conto delle dipendenze tra sistemi distribuiti, utilizzando idonei strumenti di monitoraggio e gestione.

Per il sistema nel suo complesso, tuttavia, rimodulare l'architettura in termini di container, API e integrazione significa poter prendere le decisioni più giuste per ogni servizio, punto di integrazione e interazione con il cliente. Diventa possibile, ad esempio, verificare la sicurezza e reindirizzare direttamente al servizio di back-end più appropriato un elevato numero di richieste in ingresso, senza incorrere nei colli di bottiglia creati dall'utilizzo di un singolo ESB.

Negli ambienti cloud ibridi e distribuiti, molti dei sistemi di back-end in questione possono occupare posizioni fisiche diverse. In questi casi, si ottiene più efficienza e sicurezza integrando sistemi locali, anziché reindirizzare ogni elemento attraverso un unico sistema di integrazione centralizzato che detiene la logica aziendale strategica.

## PER UNA CULTURA DELL'ORGANIZZAZIONE AGILE

Il ciclo di vita dell'infrastruttura è diverso da quello operativo o dello sviluppo del software. Quest'ultimo prevede il completamento di un progetto prima di passare al successivo; in questo caso si è efficienti se si riducono i tempi di rilascio o se aumenta il numero di funzionalità prodotte in un tempo specificato. Anche in ambito operativo, vale a dire manutenzione e stabilità, è vantaggioso poter applicare correzioni e aggiornamenti di sicurezza, distribuire nuovi servizi o ripristinare le modifiche apportate in modo rapido ed efficiente.

Nel caso dell'infrastruttura, l'approccio è differente, essendo questa oggetto di un lavoro sul lungo periodo, con team diversi e altamente specializzati che agiscono in modo diverso dai team interfunzionali impegnati in progetti di software engineering specifici. In linea di massima, i progetti di infrastruttura hanno un respiro più vasto rispetto ai progetti software, i cui cicli di rilascio brevi non sempre riescono a conseguire obiettivi o possono risultare in progetti incompleti. Come ha affermato Andrew Froelich, professionista dell'IT enterprise, su InformationWeek, l'infrastruttura presenta un punto di non ritorno, soprattutto per quel che riguarda hardware e datacenter, ma anche con il cloud pubblico: a un certo punto non è più possibile cancellare un progetto e cominciare da capo.<sup>10</sup> L'infrastruttura è permanente. Ciò non toglie che sia possibile conciliare metodologie e prestazioni dell'infrastruttura.

Evidenti per i team di sviluppo e operativi, i vantaggi dei processi reattivi e iterativi come quelli agili e DevOps lo sono meno per i team di infrastruttura. L'analisi di Froehlich sui pro e i contro dei metodi agili per l'infrastruttura tralascia un aspetto fondamentale: l'allineamento dei team di infrastruttura ai team operativi e di sviluppo. Su CIO, Rohan Pearce scrive a proposito della trasformazione dei team delle infrastrutture in cellule di lavoro dallo stile agile invece che in team funzionali.<sup>11</sup> Nei team di Telstra Enterprise Services gli sviluppatori ignoravano tout-court i sistemi interni, perché il processo di verifica dei sistemi o di inserimento delle modifiche era troppo complesso. Modificando i gruppi di lavoro, i tempi di sviluppo si sono ridotti da 212 a 42 giorni.<sup>12</sup>

Questo esempio mostra quanto sia importante modificare i processi per consentire ai team delle infrastrutture di agevolare il resto dell'organizzazione.

Le tecnologie di agile integration sostengono un'infrastruttura più agile. API, immagini dei container e integrazioni diventano oggetto di attenzione quando si parla di infrastruttura software.

---

<sup>10</sup> Andrew Froelich, "Should IT go agile? The pros and cons". 6 ottobre 2015.  
<http://www.informationweek.com/infrastructure/pc-and-servers/should-it-go-agile-the-pros-and-cons/d/d-id/1322448>

<sup>11</sup> Ronan Pearce, "Can infrastructure be agile?" 20 giugno 2013.  
[https://www.cio.com.au/article/465436/can\\_infrastructure\\_agile\\_/](https://www.cio.com.au/article/465436/can_infrastructure_agile/)

<sup>12</sup> <http://agilemanifesto.org/>

Il Manifesto Agile definisce quattro principi chiave per lo sviluppo software.<sup>12</sup> In un'infrastruttura agile basata sull'integrazione, questi principi possono essere adottati nella strategia di integrazione.

1

**Individui e interazioni hanno la priorità su processi e gli strumenti.**

Nell'ambito dell'infrastruttura, l'analisi verte sulle interazioni tra i team, che assumono forma di comunicazioni dirette, gestite da API, messaggistica e schemi di traffico; interdipendenze a livello di sistema; processi di test e rilascio, come nel caso dei flussi CI/CD.

2

**Un software efficiente prevale su una documentazione esaustiva.**

Per sua natura, l'infrastruttura deve essere funzionale 24 ore su 24, 7 giorni su 7, con un adattamento graduale da preferire a cambiamenti repentini. Pertanto, l'efficienza è sempre un requisito implicito. Una strategia dell'infrastruttura è efficiente quando le prestazioni soddisfano le aspettative dell'utente finale.

3

**La collaborazione con il cliente prevale sulla negoziazione dei contratti.**

Nei sistemi infrastrutturali, i contratti indicano ai team di infrastruttura come gestire le dipendenze di sistema, ovvero i criteri di sicurezza, gli accordi a livello di servizio e le API pubblicate. I clienti sono gli utenti, interni ed esterni, di tali sistemi. L'agilità consente loro di avere voce in capitolo e proporre eventuali modifiche da apportare ai criteri e alle interfacce associate ai sistemi, e permette di applicarle in tempi rapidi. L'adozione delle integrazioni distribuite consente di ampliare la collaborazione offrendo direttamente ai team l'autonomia e il controllo necessari per sviluppare e distribuire le integrazioni.

4

**Rispondere al cambiamento ha priorità sull'esecuzione di un piano.**

Questo principio prevede che la tecnologia supporti i processi. I sistemi dell'infrastruttura devono rimanere stabili ma le tecnologie più recenti, come i container, offrono una piattaforma con deployment singolarmente elastica, nella quale aggiungere e rimuovere dinamicamente le istanze in funzione delle richieste, automatizzare deployment e aggiornamenti, orchestrare i cambiamenti su più istanze. Le definizioni delle API pubblicate diventano strumenti riutilizzabili che rendono più coerenti le attività di sviluppo. Grazie a questo approccio, anche una piattaforma progettata per adattarsi ai cambiamenti acquisisce stabilità.

Figura 6. I principi chiave per lo sviluppo software secondo il Manifesto Agile

Nell'agile integration, la tecnologia sostiene il cambiamento culturale dei team di infrastruttura, funge da base per la strategia dell'infrastruttura, e allinea i team e le tecnologie dell'infrastruttura alle strategie aziendali e di sviluppo.

La metodologia agile identifica alcuni componenti chiave di un progetto software, come individui, build e dipendenze, e quindi definisce le relazioni tra questi elementi. Se si interpreta l'infrastruttura di integrazione come un progetto agile, si possono identificare elementi e relazioni simili e paralleli: team, immagini di container, API e punti di integrazione. La Tabella 3 descrive alcuni di questi parallelismi.

**TABELLA 3. CONFRONTO TRA ELEMENTI DEL SOFTWARE AGILE E DELL'INFRASTRUTTURA AGILE**

PROGETTO	AMBITO	DETTAGLI
Individui	Team	I team sono responsabili di alcuni specifici aspetti dell'infrastruttura. Questo elemento identifica le informazioni relative alle responsabilità dei team, ad esempio i sistemi o le API gestite dal team, i leader e gli obiettivi del team.
Moduli	API	Quando sono ben definite, le interfacce API sono stabili nel tempo, hanno roadmap proprie, sono eseguite da team specifici e forniscono una capacità specifica che offre valore all'organizzazione.
Build	Immagini del container	I rilasci si basano su unità distribuibili testate e contrassegnate, il cui deployment può essere eseguito in modo affidabile da qualsiasi team autorizzato. Sostituiscono il codice monolitico e con versioni.
Compilazione delle dipendenze	Integrazioni	Questo elemento identifica le integrazioni e le associazioni tra i diversi componenti dei sistemi distribuiti. I punti di integrazione possono essere gestiti, ordinati, ritirati, sottoposti a versioni e testati, come qualsiasi altro elemento del sistema.
Test delle build	Automazione dell'infrastruttura	Questo elemento include la gestione dell'intero ciclo, dalla capacità di testare le build software, le prestazioni e i requisiti degli utenti all'entrata in produzione e al monitoraggio di più sistemi.

### Applicazione dei principi agili alla pianificazione dell'infrastruttura

La maggior parte degli approcci alla gestione dei cambiamenti esige una documentazione completa di tutti i sottosistemi. Tale documentazione deve illustrare dettagliatamente ogni aspetto del sistema, dal metodo di monitoraggio ai parametri delle prestazioni, e specificare inoltre i team responsabili. Per aderire ai principi di agilità sono richieste collaborazione e adattabilità, e ciò è in contrasto con una gestione del cambiamento focalizzata sulla creazione di documentazione.

Invece di definire in modo prescrittivo tutte le potenziali parti coinvolte, le modifiche e i componenti di sistema, è più semplice definire un insieme di linee guida e di standard da applicare per valutare le richieste di modifica e la pianificazione. È necessario considerare i seguenti aspetti:

- Qual è l'esperienza complessiva che si desidera erogare all'utente?
- In che modo ogni elemento coinvolto, team, API e sistema, contribuisce a migliorare tale esperienza nel tempo?
- In che modo verranno definite le attività di monitoraggio e avviso necessarie per mantenere i livelli di servizio? Per quali parametri?
- Quali test automatici sono necessari per verificare il comportamento previsto?
- Qual è il flusso di rilascio che adottano i team per testare e distribuire le nuove versioni dei propri sottosistemi senza interferire con l'esperienza dell'utente?
- In che modo il malfunzionamento di un componente influisce sui livelli di servizio dell'intero sistema?

Nell'ambito dell'infrastruttura agile, la gestione dei cambiamenti va intesa più come una collaborazione continua che come un contratto vincolante.

### I pronostici sono favorevoli?

Quante probabilità di successo ha il tuo progetto? Innanzitutto, occorre conoscere i criteri con i quali viene valutato il successo: soddisfazione delle specifiche? Aumento dell'adozione da parte dei clienti? O è sufficiente rilasciare il prodotto? 4PM, un gruppo di formazione per la gestione dei progetti definisce come "Successo" l'aver completato un progetto nel pieno rispetto del budget, dei tempi previsti e delle specifiche.<sup>13</sup> Secondo questa definizione, il gruppo afferma che circa il 70% dei progetti IT non ha esito positivo.<sup>13</sup> Queste cifre però registrano un cambiamento. Una recente ricerca del Project Management Institute rivela che, rispetto ai cinque anni precedenti, una percentuale sempre più alta di progetti raggiunge gli obiettivi pianificati.<sup>14</sup> L'incremento viene imputato al maggiore allineamento tra IT e team aziendali, dal quale scaturiscono migliori informazioni sulla strategia e sulle esigenze dei clienti.<sup>8</sup>

Una delle ragioni alla base di un allineamento strategico è la creazione di team agili. La metodologia agile incoraggia di fatto la collaborazione e il feedback, una visione globale dei problemi e dei sistemi e approcci più creativi.

<sup>13</sup> 4PM.com, "Why projects fail so often". 27 settembre 2015.  
<http://4pm.com/2015/09/27/project-failure/>

<sup>14</sup> Sharon Florentine, "IT project success rates finally improving". 27 febbraio 2017.  
<https://www.cio.com/article/3174516/project-management/it-project-success-rates-finally-improving.html>

Disporre di uno stack tecnologico condiviso devia l'attenzione dall'indipendenza dal codice ai sistemi e alle rispettive interdipendenze. Ciò significa pensare a livello di sistema, considerando tutti gli elementi dell'infrastruttura software, incluso il software sviluppato internamente, i sistemi dei fornitori e le loro connessioni, come un sistema unico. Le API e i sistemi di messaggistica possono coprire l'intera infrastruttura e cooperare per unificare i sistemi software.

La possibilità di sviluppare e comprendere le API e le integrazioni distribuite nei singoli team di sviluppo o operativi, rende ancor più chiare le responsabilità dei team in merito all'integrazione. Le stesse integrazioni vengono comprese meglio, perché i team che si occupano di sviluppo e deployment riconoscono le interdipendenze tra sistemi e applicazioni.

L'adozione dell'integrazione come base dell'infrastruttura, e la conseguente distribuzione delle responsabilità tra i team crea un ambiente infrastrutturale dove gli approcci agili assumono una maggiore rilevanza.

## CONCLUSIONI: RAGGIUNGERE L'AGILE INTEGRATION

L'agilità è un processo e non un progetto.

Reagire ai cambiamenti del mercato non è mai stato così importante per le aziende. Rilasciare nuovi servizi o aggiornare quelli esistenti in tempi rapidi è una capacità conferita in larga parte dai sistemi IT. Anche reinventare l'infrastruttura IT, ovvero la base su cui poggiano i servizi digitali, non è mai stato così importante.

I team delle infrastrutture sono storicamente vincolati a lunghi processi modulati che hanno l'obiettivo di limitare i rischi e assicurare stabilità. È tuttavia giunto il momento di migrare a una tipologia di infrastruttura non più basata sull'hardware o sulla piattaforma, ma sull'integrazione. L'integrazione non è una componente dell'infrastruttura, ma un approccio concettuale all'infrastruttura che include dati e applicazioni, hardware e piattaforme.

L'agile integration definisce proprio tale approccio, ovvero un modo di usare le tecnologie di integrazione per agevolare la creazione di un'infrastruttura più agile e adattativa. L'agile integration poggia su tre pilastri:

- **Integrazione distribuita**, che adotta messaggistica e modelli di integrazione aziendale per integrare dati e sistemi, che vengono suddivisi in piccole integrazioni team-driven distribuite come necessario tra progetti e punti di contatto.
- **Gestione interna delle API**, per creare un set riutilizzabile di interfacce che consente ai team di sviluppo di interagire con applicazioni e sistemi. Le API forniscono le linee guide e la struttura dell'interazione tra le applicazioni.
- **Container**, che consentono di allineare i progetti di integrazione con quelli di sviluppo e operativi, e di sviluppare, testare e rilasciare le integrazioni come accade con i progetti software usando i metodi DevOps.

**E-BOOK** Agile integration: un modello per l'architettura aziendale

La tecnologia è spesso usata a supporto del cambiamento culturale: in questo caso opera affinché siano agili i team dell'infrastruttura, non solo il software che realizzano. Mano a mano che i team di infrastruttura si allineano ai principi di agilità, sarà possibile integrare gradualmente la tecnologia a supporto dei cambiamenti. Non esiste un unico progetto che possa rimodellare un'intera azienda per renderla agile. Può essere più efficace implementare una tecnologia di agile integration o cambiare un settore dell'attività e poi estendere in modo incrementale tali cambiamenti.

Migliorare la reattività dell'infrastruttura IT è un obiettivo di lungo termine. Per progredire, non è necessario apportare modifiche radicali né tentare cambiamenti isolati per poi propagarli all'azienda intera.

L'agile integration offre un framework tecnico e organizzativo che può aiutare l'azienda a rimodellare l'infrastruttura IT.

**INFORMAZIONI SU RED HAT**

Red Hat è il leader mondiale nella fornitura di soluzioni software open source e si avvale di un approccio community-based per offrire tecnologie cloud, Linux, middleware, storage e di virtualizzazione caratterizzate da affidabilità e prestazioni elevate. L'azienda offre inoltre servizi di supporto, formazione e consulenza per i quali ha ottenuto diversi riconoscimenti. Principale punto di riferimento in una rete globale di aziende, partner e community open source, Red Hat consente di creare tecnologie specifiche e innovative che garantiscono libero accesso alle risorse per la crescita e preparano i clienti al futuro dell'IT.



facebook.com/redhatinc  
@redhat

linkedin.com/company/red-hat

it.redhat.com  
f11423\_0518

**EUROPA, MEDIO ORIENTE****E AFRICA (EMEA)**

00800 7334 2835

it.redhat.com

europe@redhat.com

**TURCHIA**

00800-448820640

**ISRAELE**

1-809 449548

**EAU**

8000-4449549