

# 2022 ACCELERATE

## Report State of DevOps



Sponsorizzato da

## Sommario

<b>01</b>	<b>Schema riepilogativo</b>	03	
<b>02</b>	<b>Come ti collochi?</b>	08	
<b>03</b>	<b>Come puoi migliorare?</b>		
	Introduzione	19	
	Cloud	21	
	SRE e DevOps	26	
	Funzionalità tecniche DevOps	29	
	Cultura	37	
<b>04</b>	<b>Perché la sicurezza della catena di fornitura è importante</b>	42	
<b>05</b>	<b>Sorprese</b>		55
<b>06</b>	<b>Informazioni demografiche e aziendali</b>		59
<b>07</b>	<b>Considerazioni finali</b>		67
<b>08</b>	<b>Ringraziamenti</b>		68
<b>09</b>	<b>Autori</b>		69
<b>10</b>	<b>Metodologia</b>		73
<b>11</b>	<b>Per approfondire</b>		76

# 01

## Schema riepilogativo



Derek DeBellis



Claire Peters

Negli ultimi otto anni abbiamo prodotto il report Accelerate State of DevOps, ascoltando 33.000 professionisti lungo il percorso. La nostra ricerca si concentra sull'esame di come le funzionalità e le pratiche rendono prevedibili i risultati che consideriamo centrali per le DevOps:

- Prestazioni di distribuzione del software. Le quattro metriche chiave delle prestazioni di distribuzione del software: frequenza di deployment, tempo di risposta per le modifiche, tasso di errore delle modifiche e tempo per ripristinare il servizio.
- Prestazioni operative. La quinta metrica chiave, l'affidabilità.
- Prestazioni dell'organizzazione. Quanto bene la tua organizzazione soddisfa gli obiettivi di prestazioni e redditività.

Ci concentriamo anche sui fattori che sono alla base di altri risultati come il burnout e la probabilità che i dipendenti consiglino i loro team.



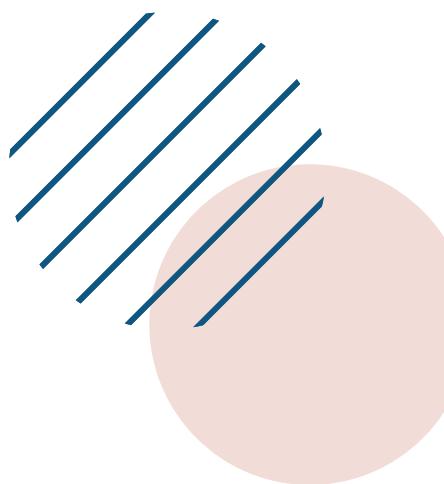
## Protezione della catena di fornitura del software

Nel 2021 abbiamo scoperto che la sicurezza della catena di fornitura del software è essenziale per raggiungere molti risultati importanti.

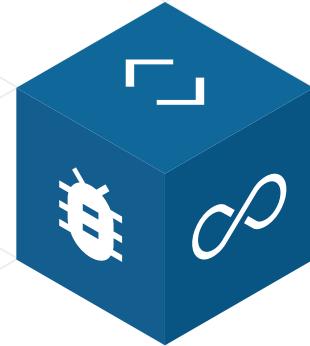
Quest'anno abbiamo approfondito la sicurezza della catena di fornitura del software, rendendola un tema centrale del nostro sondaggio e report. Abbiamo sfruttato il framework [Supply Chain Levels for Secure Artifacts \(SLSA\)](#) per esplorare prassi tecniche che supportano lo sviluppo della sicurezza della catena di fornitura del software. Abbiamo anche utilizzato il National Institute for Standards and Technology's [Secure Software Development Framework \(NIST SSDF\)](#) per esplorare atteggiamenti, processi e pratiche non tecniche relative alla protezione della catena di fornitura del software.

Abbiamo scoperto che il più grande fattore predittivo delle pratiche di sicurezza per lo sviluppo di applicazioni di un'organizzazione è culturale, non tecnico: le culture con fiducia elevata e attribuzione di colpa ridotta, incentrate sulle prestazioni, hanno una probabilità 1,6 volte maggiore di avere un'adozione superiore alla media delle pratiche di sicurezza emergenti rispetto alle culture con bassa fiducia e alta attribuzione di colpe, incentrate sul potere o sulle regole. Abbiamo anche trovato prove preliminari che suggeriscono che l'analisi della sicurezza precedente al deployment è efficace nel trovare dipendenze vulnerabili, con conseguente minor numero di vulnerabilità nel codice di produzione.

L'adozione di buone pratiche di sicurezza per lo sviluppo delle applicazioni è correlata a vantaggi aggiuntivi. Abbiamo scoperto che i team che si concentrano sulla definizione di queste pratiche di sicurezza hanno ridotto il burnout degli sviluppatori; i team con bassi livelli di pratiche di sicurezza hanno probabilità 1,4 volte maggiori di avere alti livelli di burnout rispetto ai team con alti livelli di sicurezza.<sup>1</sup> È molto più probabile che i team che si concentrano sulla definizione di pratiche di sicurezza raccomandino il proprio team a qualcun altro. Inoltre, le pratiche di sicurezza relative a SLSA predicono positivamente sia le prestazioni dell'organizzazione che le prestazioni di distribuzione del software, ma questo effetto necessita dell'applicazione di solide funzionalità di integrazione continua per emergere completamente.



<sup>1</sup> Consideriamo valori alti in questa statistica quelli con deviazione standard  $\geq 1$  sul punteggio (es. sicurezza) e bassi quelli con deviazione standard  $\leq -1$  sul punteggio.



## Driver delle prestazioni dell'organizzazione

Oltre alle pratiche di sicurezza sopra menzionate, le variabili chiave che influiscono sulle prestazioni dell'organizzazione tendono a rientrare nelle seguenti categorie:

### Cultura dell'organizzazione e dei team

Le culture con fiducia elevata e attribuzione di colpa ridotta, come definite da [Westrum](#), tendono ad avere prestazioni dell'organizzazione più elevate. Allo stesso modo, le organizzazioni con team che si sentono supportati attraverso finanziamenti e sponsorizzazioni della leadership tendono ad avere prestazioni dell'organizzazione più elevate. Anche la stabilità del team e le percezioni positive sul proprio team (probabilità di consigliare il team) tendono a portare a livelli più elevati di prestazioni dell'organizzazione. Infine, le aziende che offrono modalità di lavoro flessibili tendono a ottenere livelli elevati di prestazioni dell'organizzazione.

### Affidabilità

Sia le pratiche che associamo all'ingegneria dell'affidabilità (ad esempio obiettivi di affidabilità chiari, metriche di affidabilità salienti ecc.) sia la misura in cui le persone affermano di ritenerne soddisfatte le loro aspettative di affidabilità sono potenti predittori di alti livelli di prestazioni dell'organizzazione.

### Cloud

Abbiamo scoperto che l'utilizzo del cloud è predittivo delle prestazioni dell'organizzazione. Le aziende con software inizialmente creati su e per il cloud tendono ad avere prestazioni dell'organizzazione più elevate. Utilizzare cloud privati, cloud pubblici, cloud ibridi o una combinazione di cloud, dà origine a prestazioni dell'organizzazione più elevate rispetto all'uso dei soli server locali. Le organizzazioni che utilizzano più cloud pubblici hanno una probabilità 1,4 volte maggiore di avere prestazioni dell'organizzazione superiori alla media rispetto a coloro che non lo fanno.

L'utilizzo del cloud sembra anche avere un impatto sulle prestazioni dell'organizzazione attraverso altri fattori nel nostro set di dati. Un esempio è la sicurezza della catena di fornitura, in cui abbiamo riscontrato che le organizzazioni che utilizzano cloud pubblici hanno anche maggiori probabilità di implementare pratiche SLSA, forse perché i cloud provider incoraggiano e forniscono componenti di base per molte pratiche SLSA, come l'automazione di build e deployment.<sup>2,3</sup> Il quadro più ampio è che l'utilizzo di piattaforme cloud consente ai team di ereditare molte funzionalità e pratiche che alla fine danno come risultato prestazioni dell'organizzazione più elevate.

<sup>2</sup> Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data." Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

<sup>3</sup> Carrión, Gabriel Cepeda, Christian Nitzl, and José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples." Partial least squares path modeling. Springer, Cham, 2017. 173-195.

## Il contesto è importante

Da molto tempo, DORA tiene in considerazione il fatto che gli effetti dipendono dal contesto più ampio in cui opera il team. Riteniamo importante comprendere le caratteristiche di un team (processi, punti di forza, vincoli e obiettivi) e l'ambiente in cui si svolge il lavoro. Ad esempio, una funzionalità tecnica vantaggiosa in un contesto potrebbe essere deleteria in un altro. Quest'anno ci siamo concentrati sulla modellazione esplicita di queste condizioni ipotizzate sotto forma di interazioni; molte di queste ipotesi sono supportate dai dati di quest'anno:

- Le elevate prestazioni di distribuzione del software sono vantaggiose per le prestazioni dell'organizzazione solo quando anche le prestazioni operative sono elevate. La distribuzione rapida potrebbe non essere rilevante se il tuo servizio non è in grado di soddisfare le aspettative di affidabilità degli utenti.
- L'implementazione dei controlli di sicurezza della catena di fornitura del software, come quelli consigliati dal framework SLSA, ha un effetto positivo sulle prestazioni di distribuzione del software quando l'integrazione continua è saldamente stabilita. Senza funzionalità di integrazione continua in atto, le prestazioni di distribuzione del software e i controlli di sicurezza potrebbero essere in conflitto.
- L'impatto delle pratiche di Site Reliability Engineering (SRE) sulla capacità di un team di raggiungere gli obiettivi di affidabilità non è lineare. Praticare l'SRE non influisce positivamente sull'affidabilità fino a quando un team non raggiunge un certo livello di maturità SRE. Prima che un team raggiunga questo livello di maturità SRE, non rileviamo una relazione tra SRE e raggiungimento degli obiettivi di affidabilità. Man mano che l'adozione dell'SRE da parte di un team cresce, tuttavia, raggiunge un punto di svolta in cui l'impiego dell'SRE inizia a consentire una forte previsione dell'affidabilità. E a seguire, il miglioramento dell'affidabilità ha un impatto sulle prestazioni dell'organizzazione.
- Le funzionalità tecniche sono basate una sull'altra. La distribuzione continua e il controllo della versione amplificano le reciproche capacità di promuovere livelli elevati di prestazioni di distribuzione del software. La combinazione di distribuzione continua, architettura a basso accoppiamento, controllo della versione e integrazione continua favorisce prestazioni di distribuzione del software superiori alla somma delle parti.



Le condizioni da cui dipende la distribuzione e la necessità di comprendere il contesto più ampio di un team, ci portano a una conclusione simile a questo insight del 2021:

"Per apportare miglioramenti significativi, i team devono adottare una filosofia di miglioramento continuo. Usare i benchmark per misurare lo stato attuale, identificare i vincoli in base alle funzionalità esaminate dalla ricerca e sperimentare miglioramenti per alleviare i vincoli identificati. La sperimentazione porterà a una combinazione di vittorie e sconfitte, ma in entrambi gli scenari i team potranno adottare azioni significative in conseguenza delle lezioni apprese."

In effetti, quest'anno abbiamo riscontrato un effetto profondamente allineato a questa filosofia generale: i team che riconoscono la necessità di migliorare continuamente tendono ad avere prestazioni dell'organizzazione più elevate rispetto a quelli che non lo fanno.

In breve, i team devono adattarsi continuamente e sperimentare le pratiche di sviluppo del software.

Lo sappiamo perché, nel complesso, i team che fanno questo hanno prestazioni dell'organizzazione più elevate. Non sempre (ciò che funziona per un'organizzazione non necessariamente funziona per un'altra), ma la maggior parte delle volte. Mentre ti impegni nei tuoi esperimenti con le pratiche DevOps, preparati al fallimento occasionale mentre ti perfezioni su ciò che funziona per il tuo team.

Quest'anno abbiamo anche avuto una serie di [sorprese](#) nei dati, ma dovrai continuare a leggere per scoprire quali sono.

I team che riconoscono la necessità di migliorare continuamente tendono ad avere prestazioni dell'organizzazione più elevate rispetto a quelli che non lo fanno.

# 02

## Come ti collochi?



Derek DeBellis

Vuoi sapere dove si colloca il tuo team rispetto ad altri nel settore? Questa sezione include la più recente valutazione benchmark delle prestazioni DevOps. Esaminiamo in che modo i team sviluppano, distribuiscono ed eseguono i sistemi software e suddividiamo quindi gli intervistati in cluster che inquadrano le combinazioni più comuni delle prestazioni DevOps.

Quest'anno abbiamo incluso due diversi approcci di clustering. Il primo si basa su precedenti storici. Questo approccio di clustering si concentra sulla creazione di cluster basati su quattro metriche che inquadrano le prestazioni di distribuzione del software: tempo di risposta, frequenza di distribuzione, tempo per ripristinare il servizio e tasso di errore delle modifiche. Riassumeremo ciascuna di queste di seguito. L'obiettivo di questo approccio è aiutarti a quantificare le prestazioni attuali del tuo team in modo da poter confrontare le vostre prestazioni con quelle di altri team.

Il secondo approccio di clustering include una quinta metrica, l'affidabilità, che utilizziamo per comprendere le prestazioni operative. Perché aggiungere una nuova metrica alla cluster analysis? Perché abbiamo costantemente visto l'importanza di questa metrica. In effetti, abbiamo prove che suggeriscono che le prestazioni di distribuzione possono essere dannose per le prestazioni dell'organizzazione se non abbinate a forti prestazioni operative. A differenza del nostro approccio di clustering tradizionale, questo è un esercizio descrittivo che tenta di tracciare un quadro dei risultati comuni dei team per quanto riguarda sia le prestazioni di distribuzione che quelle operative. Quindi, non è sempre ovvio quale cluster sia migliore.

Innanzitutto, diamo un'occhiata a una breve panoramica delle cinque misure che stiamo utilizzando per comprendere la distribuzione del software e le prestazioni operative.



## Prestazioni di distribuzione del software e operative

Per soddisfare le esigenze dei loro settori in continua evoluzione, le organizzazioni devono fornire e utilizzare il software in modo rapido e affidabile. Maggiore è la velocità con cui i tuoi team possono apportare modifiche al software, minore sarà il tempo necessario a fornire valore ai tuoi clienti, eseguire esperimenti e ricevere feedback preziosi. Con otto anni di raccolta di dati e ricerca alle spalle, abbiamo sviluppato e convalidato quattro metriche che misurano le prestazioni di distribuzione del software. A partire dal 2018, abbiamo incluso una quinta metrifica

volta a inquadrare le capacità operative. I team che eccellono in tutte e cinque le misure mostrano prestazioni dell'organizzazione eccezionali. Chiamiamo queste cinque misure **prestazioni di distribuzione del software e operative (SDO, Software Delivery and Operational)**. Tieni presente che queste metriche si concentrano sui risultati a livello di sistema, il che aiuta a evitare le insidie più frequenti del monitoraggio delle metriche del software, che possono comportare il conflitto tra le funzioni e la realizzazione di ottimizzazioni locali a scapito dei risultati complessivi.



## Cinque metriche per le prestazioni di distribuzione e operative

Le quattro metriche delle prestazioni di distribuzione del software possono essere considerate in termini di velocità effettiva e stabilità. Misuriamo la velocità effettiva utilizzando il **tempo di risposta delle modifiche al codice** (ovvero il tempo dal commit del codice al rilascio in produzione) e la **frequenza di deployment**. Misuriamo la stabilità utilizzando il **tempo per ripristinare un servizio** dopo un incidente e il **tasso di errore delle modifiche**.

La quinta metrica rappresenta le **prestazioni operative** ed è una misura delle pratiche operative moderne. Basiamo le **prestazioni operative** sull'**affidabilità**, ovvero quanto i tuoi servizi soddisfano le aspettative degli utenti, come disponibilità e prestazioni. Storicamente abbiamo misurato la disponibilità piuttosto che l'affidabilità, ma poiché la disponibilità è un focus specifico dell'ingegneria dell'affidabilità, nel 2021 abbiamo voluto estendere la misurazione all'affidabilità, in modo che disponibilità, latenza, prestazioni e scalabilità fossero rappresentate in modo più ampio. Nello specifico, abbiamo chiesto agli intervistati di valutare la loro capacità di raggiungere o superare gli obiettivi di affidabilità. Abbiamo scoperto che i team con diversi gradi di prestazioni di distribuzione ottengono risultati migliori (ad es. meno burnout) quando danno priorità anche alle prestazioni operative.

Prestazioni di distribuzione  
del software



Tempo di risposta  
per le modifiche



MTTR



Frequenza di  
deployment

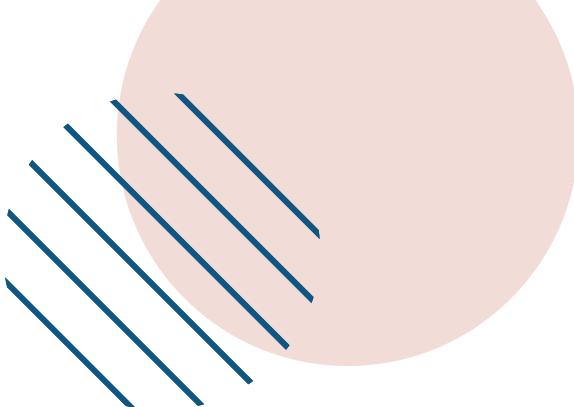


Tasso di errore  
delle modifiche

Prestazioni  
operative



Affidabilità



## Approccio al clustering storico: clustering delle prestazioni di distribuzione

Quest'anno, durante la valutazione della soluzione a quattro cluster che utilizziamo dal 2018, abbiamo notato che i dati mostravano un chiaro cluster con basse prestazioni e un chiaro cluster con alte prestazioni. Tuttavia, i due cluster che tradizionalmente utilizzeremmo per delimitare prestazioni medie e prestazioni elevate non erano sufficientemente differenziati da giustificare una divisione. Inoltre, i vari indici che utilizziamo per scegliere la giusta soluzione

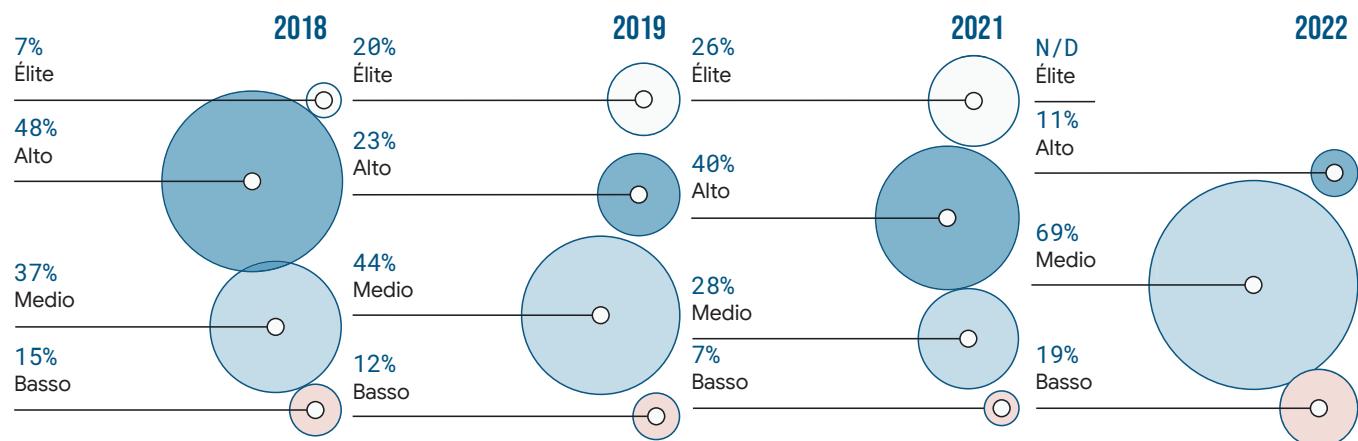
di cluster invariabilmente suggerivano che tre cluster catturassero al meglio i dati, indipendentemente dalle tecniche di clustering applicate. La tabella seguente descrive le caratteristiche delle prestazioni di distribuzione per ciascun cluster.

La differenza notevole rispetto allo scorso anno è che quest'anno non consideriamo nessun cluster un'élite. Il cluster alto di quest'anno è una miscela del cluster alto e di quello élite dell'anno scorso. Abbiamo deciso di omettere un cluster élite perché il cluster con le prestazioni più elevate semplicemente non indica abbastanza delle caratteristiche del cluster élite dell'anno scorso.

Metrica per le prestazioni di distribuzione del software	Basse	Medie	Alte
<b>Frequenza di deployment</b> Per l'applicazione o il servizio principale su cui lavori, con quale frequenza la tua organizzazione esegue il deployment del codice in produzione o il rilascio agli utenti finali?	Tra una volta al mese e una volta ogni sei mesi	Tra una volta a settimana e una volta al mese	On demand (più deployment al giorno)
<b>Tempo di risposta per le modifiche</b> Per l'applicazione o il servizio principale su cui lavori, qual è il tuo tempo di risposta per le modifiche, ovvero quanto tempo serve per passare dal commit del codice alla sua corretta esecuzione in produzione?	Tra un mese e sei mesi	Tra una settimana e un mese	Tra un giorno e una settimana
<b>Tempo per ripristinare il servizio</b> Per l'applicazione o il servizio principale su cui lavori, quanto tempo serve in genere per ripristinare il servizio quando si verifica un incidente del servizio o un difetto che impatta sugli utenti, ad esempio un'interruzione non pianificata o una compromissione del servizio?	Tra una settimana e un mese	Tra un giorno e una settimana	Meno di un giorno
<b>Tasso di errore delle modifiche</b> Per l'applicazione o il servizio principale su cui lavori, che percentuale delle modifiche alla produzione o rilasciate agli utenti finali provoca una riduzione del servizio (ad esempio, portando a una compromissione o un'interruzione del servizio) e richiede di conseguenza una risoluzione (ad esempio, l'applicazione di hotfix, rollback, fix forward, patch)?	46%-60%	16%-30%	0%-15%

Suggerisce che questo campione non rappresenta team o organizzazioni con dipendenti che ritengono di aver fatto rapidi progressi. Una possibile ipotesi, che ancora non è supportata da dati sufficienti, è che lo sviluppo del software abbia visto una ridotta innovazione in termini di pratiche, strumenti e condivisione delle informazioni. Questo potrebbe essere il risultato della pandemia in corso, che ostacola la capacità di condividere conoscenze e pratiche tra team e organizzazioni. Potrebbero esserci state meno opportunità di riunirsi e imparare gli uni dagli altri, il che, a sua volta, potrebbe aver rallentato l'innovazione. Speriamo di approfondire l'argomento per esplorare ciò che è alla base di questa scoperta.

Detto questo, se confronti i cluster basso, medio e alto di quest'anno con quelli dell'anno scorso, vedrai che c'è uno spostamento verso prestazioni di distribuzione leggermente superiori. Sembra che i cluster di quest'anno siano compresi tra due di quelli dell'anno scorso. Il cluster alto del 2022 è compreso tra il cluster alto e quello élite del 2021. Il cluster basso del 2022 sembra essere compreso tra i cluster basso e medio del 2021. Lo spostamento verso l'alto per il cluster a basse prestazioni suggerisce che mentre il tetto per le prestazioni di distribuzione è stato abbassato, il livello minimo è stato alzato.

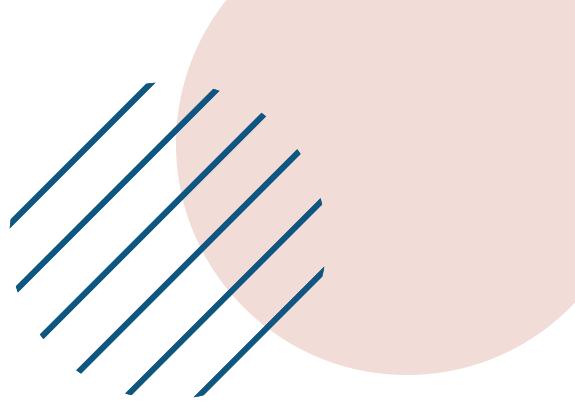


Le scomposizioni percentuali nella tabella sopra, mostrano che la percentuale di intervistati con prestazioni elevate è al livello minimo degli ultimi 4 anni, mentre la percentuale di intervistati con prestazioni basse è aumentata notevolmente, dal 7% del 2021 al 19% di quest'anno. Oltre i due terzi degli intervistati di quest'anno rientrano nel cluster medio. Il netto calo di intervistati con prestazioni elevate ed élite potrebbe suggerire che molti degli intervistati di quest'anno fanno parte di organizzazioni o team che non hanno stabilito, né sono in procinto di stabilire, la cultura DevOps che stiamo vedendo emergere in molti team moderni.

Forse ci stiamo concentrando troppo sulle differenze tra il 2021 e il 2022, piuttosto che evidenziare le somiglianze. I cluster del 2021 e del 2022 condividono molte caratteristiche, inclusa un'enorme separazione tra i titoli con alte prestazioni e quelli a con basse prestazioni. Ad esempio, si stima che gli intervistati con prestazioni elevate<sup>1</sup> abbiano 417 volte più deployment rispetto agli intervistati con prestazioni basse.

<sup>1</sup>Vedi la sezione "Metodologia" per capire cosa c'è dietro questa stima





## Clustering delle prestazioni di distribuzione e delle prestazioni operative

Abbiamo deciso di fare un'analisi del cluster sulle tre categorie che le cinque metriche sono progettate per rappresentare: **velocità effettiva** (un composto di tempo di risposta delle modifiche al codice e frequenza di deployment), **stabilità** (un composto di tempo per ripristinare il servizio e tasso di errore delle modifiche) e **prestazioni operative** (affidabilità). La ragione di ciò è il ruolo fondamentale svolto dalle prestazioni

operative nei nostri modelli. Per le organizzazioni che non mostrano prestazioni operative solide, la velocità effettiva e la stabilità hanno un impatto minore sulle prestazioni dell'organizzazione.

Riteniamo che una descrizione del panorama delle prestazioni DevOps che non tenga conto delle prestazioni operative tralascia una parte cruciale del quadro.

L'esplorazione dei dati ci ha portato a una soluzione a quattro cluster. Ecco una ripartizione dei quattro cluster e dei loro nomi:

Cluster	Stabilità		Prestazioni operative	Velocità effettiva		% degli intervistati
	Tempo per ripristinare il servizio	Tasso di errore delle modifiche	Affidabilità	Tempo di risposta	Frequenza di deployment	
In fase di avvio	Tra un giorno e una settimana	31%-45%	A volte soddisfa le aspettative	Tra una settimana e un mese	Tra una volta a settimana e una volta al mese	28%
In corso	Meno di un'ora	0%-15%	Generalmente soddisfa le aspettative	Meno di un giorno	On demand (più deployment al giorno)	17%
In fase di rallentamento	Meno di un giorno	0%-15%	Generalmente soddisfa le aspettative	Tra una settimana e un mese	Tra una volta a settimana e una volta al mese	34%
In fase di ritiro	Tra un mese e sei mesi	46%-60%	Generalmente soddisfa le aspettative	Tra un mese e sei mesi	Tra una volta al mese e una volta ogni sei mesi	21%

Tuttavia, se visitassi due team nello stesso cluster, probabilmente sembrerebbero molto diversi e la nostra storia potrebbe non rappresentare ciò che vedi. La storia che presentiamo per ogni cluster qui sopra è un tentativo di utilizzare la nostra esperienza di lavoro con molti team per rendere intelligibili questi modelli nei dati. Inoltre, se visitassi lo stesso team in due momenti diversi, potrebbe non essere ancora nello stesso gruppo. Una ragione potrebbe essere che il team è migliorato o peggiorato; un'altra possibilità è che il team sia passato a un modello di deployment più appropriato per lo stato corrente della sua applicazione o del suo servizio. Ad esempio, in una fase precedente dello sviluppo di un'applicazione o di un servizio, un team potrebbe essere stato concentrato sull'esplorazione (cluster In fase di avvio), ma quando iniziano a trovare la propria nicchia potrebbero spostare l'attenzione sull'affidabilità (cluster In corso o In fase di rallentamento).

## Ogni cluster ha caratteristiche uniche e rappresenta una parte sostanziale delle risposte.

Il cluster **In fase di avvio** non presenta prestazioni buone né scadenti in nessuna delle nostre dimensioni. Gli intervistati di questo cluster potrebbero trovarsi nelle prime fasi dello sviluppo del prodotto, della funzionalità o del servizio. Potrebbero essere meno focalizzati sull'affidabilità perché si stanno concentrando sull'ottenere feedback, sul comprendere l'adattamento del loro prodotto al mercato e, più in generale, sull'esplorazione.

Il cluster **In corso** si comporta bene in tutte le caratteristiche: elevata affidabilità, elevata stabilità, elevata velocità effettiva. Solo il 17% degli intervistati si trova nello stato In corso.

Gli intervistati del cluster **In fase di rallentamento** non eseguono il deployment troppo spesso, ma quando lo fanno è probabile che abbiano successo. Oltre un terzo delle risposte rientra in questo cluster, rendendolo il più ampio e rappresentativo del nostro campione. Questo modello è probabilmente tipico (sebbene tutt'altro che esclusivo) dei team che stanno migliorando in modo incrementale, ma loro e i loro clienti sono per lo più soddisfatti dello stato attuale della loro applicazione o prodotto.

Infine, il cluster **In fase di ritiro** sembra per i team che stanno lavorando su un servizio o un'applicazione ancora preziosa per loro e per i loro clienti, ma non più in fase di sviluppo attivo.



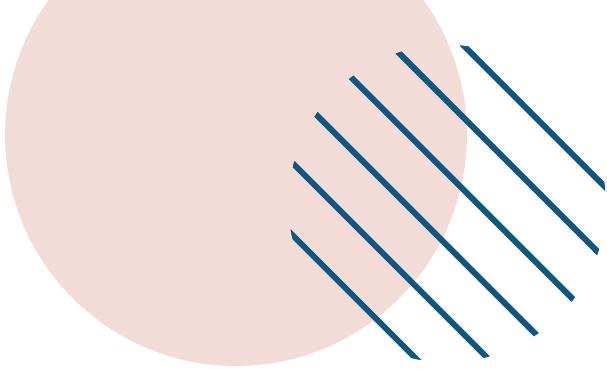
Questi cluster differiscono in particolare nelle loro pratiche e funzionalità tecniche. Dati gli alti livelli delle prestazioni di distribuzione del software e operative dimostrati dagli intervistati del cluster **In corso**, abbiamo deciso di esaminare in che modo si differenziano dagli intervistati degli altri cluster nelle pratiche e nelle funzionalità tecniche. Abbiamo scoperto che rispetto agli altri cluster, il cluster **In corso** si concentra maggiormente su:

- Architetture a basso accoppiamento: la misura in cui i team possono apportare modifiche su larga scala alla progettazione del proprio sistema senza dipendere dalla modifica dei sistemi di altri team
- Fornire flessibilità: quanto è flessibile un'azienda per quanto riguarda le modalità di lavoro dei dipendenti
- Controllo della versione: come vengono gestite le modifiche al codice dell'applicazione, alla configurazione del sistema, alla configurazione dell'applicazione ecc.
- Integrazione continua (CI): frequenza con cui i rami vengono integrati nel tronco
- Distribuzione continua (CD): funzionalità incentrate sull'inserimento delle modifiche in produzione in modo sicuro, sostenibile ed efficiente

Curiosamente, il cluster **In corso** tende a concentrarsi meno sulla documentazione. L'anno scorso, abbiamo riscontrato che le pratiche di documentazione sono fondamentali sia per le prestazioni di distribuzione che per le prestazioni operative (SDO). In che modo il cluster **In corso** ha prestazioni SDO elevate senza una forte attenzione alla documentazione? Per prima cosa, ci sono molti percorsi per ottenere prestazioni SDO elevate al di fuori della documentazione. Inoltre, forse il cluster **In corso** effettua continuamente il refactoring del proprio codice per creare un processo più auto-documentante e, quindi, avere meno bisogno di documenti così come li descriviamo nel sondaggio.

Il cluster **In fase di rallentamento**, che costituisce la percentuale più alta dei nostri intervistati, tende a essere composto da intervistati di organizzazioni più grandi che tendono a essere meno cloud-native rispetto a quelli di altri cluster. Immagina un'azienda molto matura con alcuni processi calcificati che, alla fine della giornata, forniscono ancora agli utenti finali esperienze stabili e affidabili (e di valore). Questo cluster mostra una cultura generativa orientata alle prestazioni.<sup>2</sup> Una delle caratteristiche più interessanti del cluster In fase di rallentamento è che ha una velocità effettiva bassa e una cultura del lavoro positiva (cultura del lavoro "generativa", secondo Westrum) alta; questa è una combinazione non comune. È più comune vedere una velocità effettiva e una cultura del lavoro proporzionali (alto/alto o basso/basso). Ci auguriamo di condurre ricerche future per comprendere meglio la relazione tra velocità effettiva e cultura.

<sup>2</sup> (Westrum)



Abbiamo anche esaminato il confronto tra questi diversi cluster su tre risultati: burnout, prestazioni dell'organizzazione e lavoro non pianificato. Quello che abbiamo scoperto ha infranto le nostre aspettative.

Il cluster **In fase di ritiro** ha avuto prestazioni dell'organizzazione superiori rispetto agli altri cluster. Osservando le caratteristiche di questo cluster (scarsa stabilità e scarsa velocità effettiva) questo è apparentemente in contrasto con la maggior parte delle precedenti scoperte di DORA. Ma invece di incolpare la casualità per aver creato un'anomalia (altamente possibile), vogliamo esplorare alcune possibili spiegazioni.

Quando si cerca di decomprimere questi risultati, c'è un'importante scoperta complementare da tenere a mente. Il cluster **In fase di ritiro** raggiunge prestazioni dell'organizzazione più elevate a un costo non indifferente: i suoi team hanno i tassi di burnout più alti, si sentono più soggetti a errori e sono gravati dal carico maggiore di lavoro non pianificato. Insieme, questi risultati suggeriscono che l'affidabilità potrebbe essere sufficiente per ottenere prestazioni dell'organizzazione elevate, ma senza velocità e stabilità, il tuo team pagherà il costo del burnout e del lavoro non pianificato.

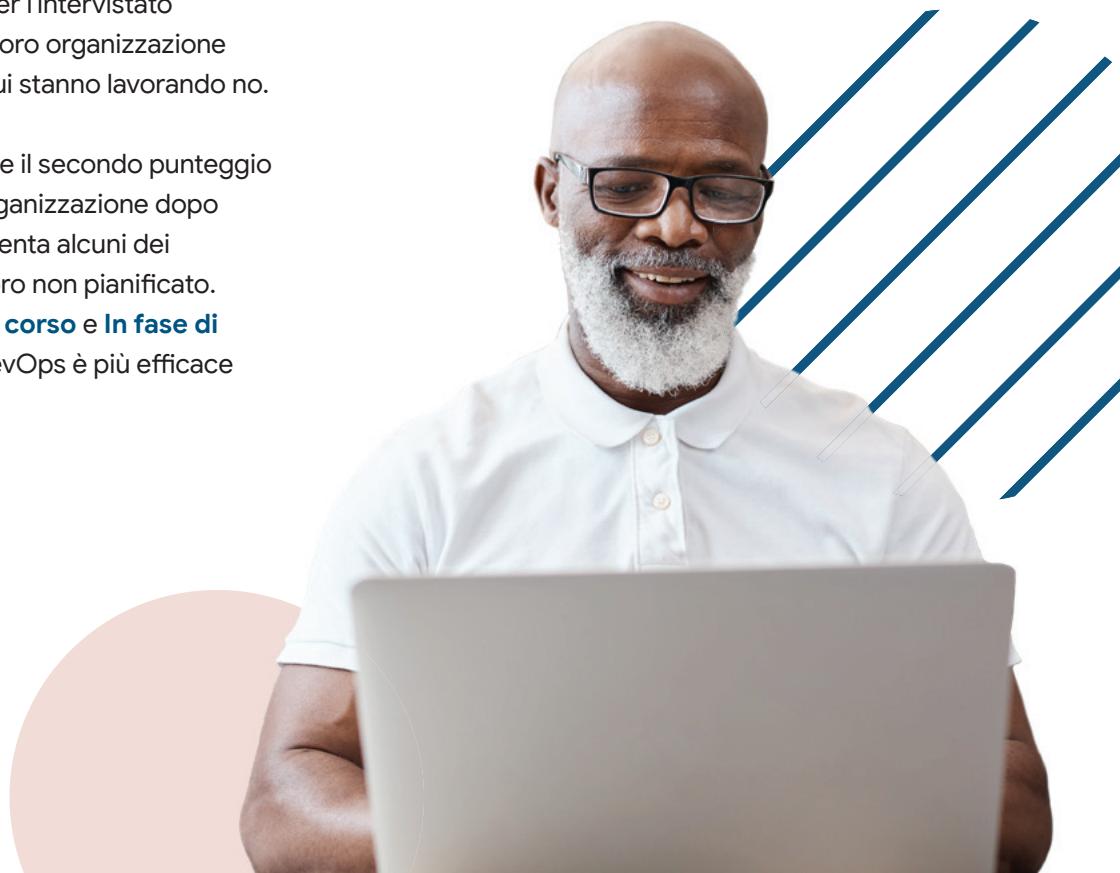
Abbiamo altre ipotesi per spiegare perché il cluster **In fase di ritiro** ha prestazioni dell'organizzazione più elevate rispetto agli altri cluster, in particolare il cluster **In corso**. Ecco un breve elenco.

- Esistono caratteristiche alla base di questi quattro cluster che potrebbero non essere presenti nei nostri dati. Ad esempio, le dimensioni dell'organizzazione possono essere un buon indicatore della maturità. Il cluster **In corso** tende a provenire da aziende più piccole, il che potrebbe indicare che i loro prodotti sono in fasi più formative.
- I membri del cluster **In corso** tendono a provenire da aziende più piccole, che potrebbero essere meno vincolate da processi e infrastrutture storici e, di conseguenza, disporre di processi DevOps più sofisticati. Detto questo, i dati mostrano che le dimensioni di un'organizzazione sono positivamente correlate alle prestazioni dell'organizzazione per motivi che potrebbero essere in gran parte estranei alla tecnologia.
- Il cluster **In corso** tendeva a ignorare i principi descritti nella cultura generativa di Westrum. Abbiamo visto che questo è spesso a scapito delle prestazioni dell'organizzazione.
- Le organizzazioni in ciascun cluster possono differire su cosa intendono per aspettative di affidabilità e su come le monitorano. Lo stesso vale per il modo in cui definiscono i loro obiettivi per le prestazioni dell'organizzazione.

- Il cluster **In fase di ritiro** può avere prestazioni dell'organizzazione elevate a breve termine, ma ci chiediamo come si comporterebbe a lungo termine. Il burnout si tradurrà in turnover? Saranno in grado di scalare i propri processi?
- Stiamo ponendo le domande a vari livelli. Le domande sulle funzionalità tecniche (ad esempio, l'architettura a basso accoppiamento) sono poste a livello di team, quelle sulle prestazioni dell'organizzazione sono poste a livello di organizzazione. Le organizzazioni hanno spesso molti team ed è possibile per l'intervistato riconoscere che mentre la loro organizzazione funziona bene, il team su cui stanno lavorando no.

Inoltre, il cluster **In corso** ottiene il secondo punteggio più alto nelle prestazioni dell'organizzazione dopo il cluster **In fase di ritiro** e presenta alcuni dei livelli più bassi di burnout e lavoro non pianificato. Come dimostrato dai cluster **In corso** e **In fase di rallentamento**, una filosofia DevOps è più efficace quando è presente l'affidabilità.

Siamo entusiasti di continuare a esplorare nuovi modi per descrivere le variazioni all'interno del settore. Andando avanti, vogliamo continuare a includere le prestazioni operative come una dimensione rilevante per la nostra comprensione di queste variazioni. Vogliamo anche evitare i cluster altamente prescrittivi e valutativi (ad esempio **Elite**) e concentrarci invece sull'esercizio semplicemente descrittivo di identificare costellazioni comuni di prestazioni SDO.



# 03

## Come puoi migliorare?



Derek DeBellis

### Come si migliora attraverso una moltitudine di risultati?

Il report State of DevOps è progettato per fornire una guida basata su prove per aiutare il tuo team a concentrarsi sulle pratiche e sulle funzionalità DevOps che consentono di ottenere i risultati che ti interessano. Quest'anno abbiamo ampliato la nostra indagine sia sulla sicurezza che sul set di risultati desiderati dai team. In passato, ci siamo concentrati sui risultati relativi alle prestazioni operative e di distribuzione del software (SDO) e alle prestazioni dell'organizzazione. Lo facciamo ancora, ma volevamo anche esplorare il burnout, la probabilità di consigliare il team, il lavoro non pianificato e la propensione all'errore, non solo come mezzo per migliorare le prestazioni SDO e dell'organizzazione, ma anche come indicatori a sé stanti. Di conseguenza, quest'anno, ci siamo assicurati di menzionare pratiche e funzionalità che sembrano esercitare un'influenza su questi risultati.





Il modello di ricerca è cambiato quest'anno per riflettere meglio una teoria alla base di DORA: non esiste un approccio unico per le DevOps. In pratica, abbiamo scoperto che per formulare suggerimenti è necessario comprendere il contesto più ampio di un team. Una pratica vantaggiosa per un team potrebbe essere dannosa per un altro. Ad esempio, abbiamo a lungo ipotizzato che le funzionalità tecniche (come l'architettura a basso accoppiamento, lo sviluppo basato su trunk, il controllo della versione e l'integrazione continua) abbiano un impatto positivo più pronunciato sulle prestazioni di distribuzione del software quando è attiva la distribuzione continua. Quest'anno abbiamo modellato esplicitamente questa e altre interazioni. L'obiettivo è migliorare la nostra comprensione da semplicemente "cosa ha effetto su cosa?" a includere "in quali condizioni questi effetti esistono, vengono amplificati o attenuati?" Comprendere tutta questa condizionalità si è rivelata un'impresa complicata e rivoluzionaria, ma siamo entusiasti di condividere con voi alcune di queste prime scoperte.

Puoi trovare i modelli della ricerca di quest'anno e degli anni precedenti online sul [nostro sito web](#).

## Oltre le quattro chiavi

In che modo le metriche DORA migliorano le prestazioni di sviluppo e operative? Un team interfunzionale di ingegneri del software presso Liberty Mutual Insurance esamina regolarmente le prestazioni utilizzando le [metriche delle "quattro chiavi" di DORA](#). Ad esempio, Jenna Dailey, Sr. Scrum Master presso Liberty Mutual, ha raccontato che una squadra ha sfruttato la ricerca di DORA per aiutare il team a identificare un collo di bottiglia, passare a un approccio di sviluppo basato sui test e realizzare miglioramenti nelle prestazioni complessive.

Scopri di più sull'approccio di Liberty Mutual allo sfruttamento dei dati e delle metriche DORA per migliorare la qualità e la distribuzione del software nei loro recenti [Tomorrow Talk](#).



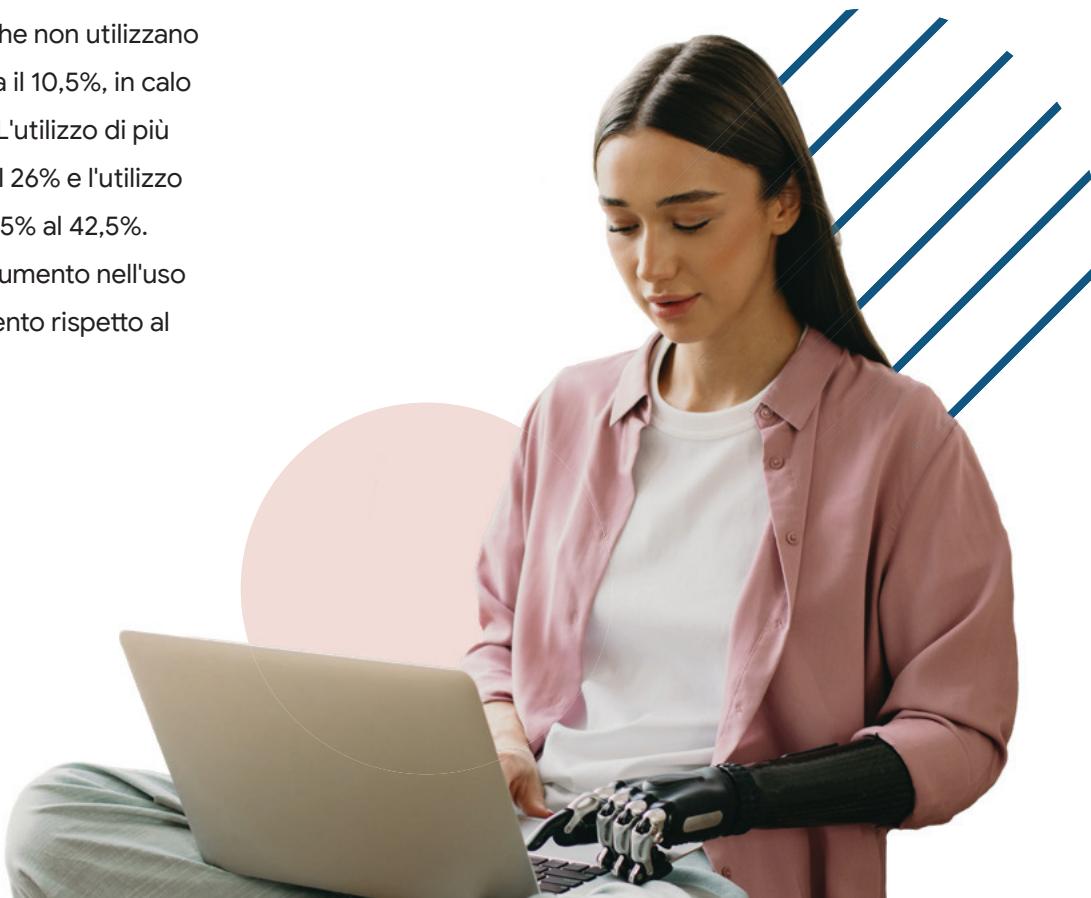
Eric Maxwell

## Cloud

Basandosi sullo slancio che abbiamo visto negli ultimi anni, l'uso del cloud computing continua ad accelerare. Infatti, la percentuale di persone che segnalano l'uso del cloud pubblico, inclusi più cloud, è ora del 76%, rispetto al 56% del 2021. Il numero di persone che riferiscono di non utilizzare affatto il cloud, comprese quelle che non utilizzano il cloud privato, è sceso ad appena il 10,5%, in calo rispetto al 21% dello scorso anno. L'utilizzo di più cloud pubblici è passato dal 21% al 26% e l'utilizzo del cloud ibrido è aumentato dal 25% al 42,5%. Abbiamo anche visto un piccolo aumento nell'uso del cloud privato al 32,5%, in aumento rispetto al 29% dello scorso anno.

Aumento del  
**36%**  
nell'utilizzo del  
cloud pubblico

Aumento del  
**25%**  
nell'utilizzo del  
cloud ibrido



L'uso del cloud computing ha un impatto positivo sulle prestazioni complessive dell'organizzazione. Gli intervistati che utilizzano il cloud hanno il **14% di probabilità in più di superare** gli obiettivi di prestazione dell'organizzazione rispetto ai colleghi che non utilizzano il cloud.

Come abbiamo dimostrato negli anni precedenti e continuiamo a riscontrare in questo report, l'uso del cloud computing ha un impatto positivo sulle prestazioni complessive dell'organizzazione. Gli intervistati che utilizzano il cloud hanno il **14% di probabilità in più di superare** gli obiettivi di prestazione dell'organizzazione rispetto ai colleghi che non utilizzano il cloud. La nostra ricerca mostra che il cloud computing consente ai team di eccellere in cose come la sicurezza e l'affidabilità della catena di fornitura del software e queste cose portano alle prestazioni dell'organizzazione.

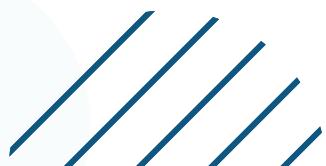
Sorprendentemente, gli utenti di tutti i tipi di cloud (pubblico, privato, ibrido e multi-cloud) hanno mostrato un'associazione negativa con il tasso di errore delle modifiche, il che significa un aumento del tasso

di errore delle modifiche. Questo merita ulteriori indagini. Invece di speculare sulle ragioni di ciò, indagheremo ulteriormente in ricerche future. Ma con poche eccezioni, l'uso di applicazioni cloud-native (applicazioni originariamente ideate e progettate per il cloud) si è distinto con segnali positivi in tutti i parametri su cui verteva il sondaggio.

L'uso di qualsiasi piattaforma di cloud computing, pubblica o privata, contribuisce positivamente ai risultati riguardanti la cultura e l'ambiente di lavoro (ad esempio, cultura generativa, minor burnout, maggiore stabilità e maggiore soddisfazione dei dipendenti). Gli intervistati che utilizzano il cloud hanno ottenuto un punteggio superiore del 16% per questi risultati culturali.

## L'utilizzo del cloud continua ad accelerare su base annua

	2022	% di variazione rispetto al 2021
<b>Cloud ibrido</b>	42,47%	25%
<b>Cloud pubblico o più cloud pubblici</b>	76,08%	36%
<b>Cloud privato</b>	32,55%	12%
<b>Nessun cloud</b>	10,55%	-50%



L'uso dell'ibrido e del multi-cloud (e privato) sembra avere un impatto **negativo** sugli indicatori delle prestazioni di distribuzione del software – MTTR, tempo di risposta e frequenza di deployment – a meno che gli intervistati non avessero alti livelli di **affidabilità**.

## Il cloud ibrido e il multi-cloud migliorano le prestazioni dell'organizzazione

Continuiamo a vedere forti segnali di come l'uso del cloud ibrido e di più cloud pubblici abbia un impatto positivo sulle organizzazioni. I professionisti che utilizzano più cloud hanno mostrato prestazioni dell'organizzazione 1,4 volte migliori rispetto a quelli che non utilizzano il cloud. Tuttavia, l'uso dell'ibrido e del multi-cloud (così come del cloud privato) sembra avere un impatto negativo su diversi indicatori delle prestazioni di distribuzione del software (MTTR, tempo di risposta e frequenza di deployment) a meno che gli intervistati non abbiano anche alti livelli di affidabilità.

Questa scoperta sottolinea ulteriormente l'importanza di una solida pratica SRE e del ruolo che l'affidabilità gioca nella distribuzione del software.

Nel 2021 abbiamo chiesto agli intervistati di dirci il motivo *principale* per cui utilizzavano più cloud pubblici, mentre nel 2022 abbiamo chiesto ai partecipanti di dirci tutti i vantaggi che ottengono dall'utilizzo di più cloud provider. La disponibilità è stato il vantaggio più citato in assoluto, il che coincide con l'attenzione per l'affidabilità che abbiamo visto nel settore: non puoi avere servizi affidabili se non sono disponibili. Oltre il 50% dei professionisti ha riferito di sfruttare i vantaggi unici di diversi cloud provider.

## Vantaggi ottenuti adottando più cloud provider

Disponibilità	62,61%
Sfruttare i vantaggi unici di ciascun provider	51,59%
La fiducia è suddivisa tra più provider	47,54%
Ripristino di emergenza	43,48%
Aspetti legali e conformità	37,97%
Tattica di negoziazione o requisito di approvvigionamento	19,13%
Altro	4,06%



degli intervistati ha riferito di utilizzare più cloud provider.

L'uso delle cinque caratteristiche del cloud computing è un inizio cruciale per una lunga catena causale che porta alle prestazioni dell'organizzazione.

## Le cinque caratteristiche del cloud computing

In linea con il nostro precedente approccio di ricerca, abbiamo cercato non solo di sapere se i partecipanti utilizzano tecnologie di cloud computing, ma come le utilizzano. Abbiamo raggiunto questo obiettivo chiedendo informazioni sulle cinque caratteristiche essenziali del cloud computing, come definito dal National Institute of Standards and Technology (NIST).

**Self-service on demand:** i consumatori possono eseguire il provisioning delle risorse di calcolo secondo necessità, automaticamente, senza che sia necessaria alcuna interazione umana da parte del provider.

**Ampio accesso alla rete:** le funzionalità sono ampiamente disponibili e i consumatori possono accedervi tramite più client come telefoni cellulari, tablet, laptop e workstation.

**Pooling di risorse:** le risorse del provider sono riunite in pool in un modello multi-tenant e le risorse fisiche e virtuali vengono assegnate e riassegnate dinamicamente on demand.

In genere il cliente non ha un controllo diretto sulla località esatta delle risorse fornite, ma può specificarla a un livello di astrazione superiore, come paese, stato o data center.

**Elasticità rapida:** l'elasticità nel provisioning e nel rilascio delle risorse consente di fare rapidamente lo scale in o lo scale out in base alla domanda. Le capacità che il consumatore ha a disposizione per il provisioning appaiono illimitate e possono essere richieste in ogni quantità, in qualunque momento.

**Servizio misurato:** i sistemi cloud controllano e ottimizzano automaticamente l'uso delle risorse sfruttando una capacità di misurazione a un livello di astrazione adeguato al tipo di servizio, come spazio di archiviazione, elaborazione, larghezza di banda e account utente attivi. L'uso delle risorse può essere monitorato, controllato e inserito in report per scopi di trasparenza.

Questo report convalida i precedenti tre anni di ricerca di DORA, concludendo che la presenza di queste cinque caratteristiche in un'organizzazione influisce positivamente sulla distribuzione del software e sulle prestazioni operative. Abbiamo anche scoperto che queste caratteristiche portano a migliori prestazioni dell'organizzazione mettendo in moto processi che influiscono positivamente su quest'ultima. Esportare le cinque caratteristiche del cloud computing è il primo passo di un lungo viaggio che porta a prestazioni dell'organizzazione più elevate.

Nel 2022 vediamo che i team stanno mettendo a frutto sempre di più i differenziatori del cloud computing. Per il quarto anno consecutivo, stiamo assistendo a una crescente adozione delle cinque caratteristiche del cloud computing. Il pooling delle risorse ha registrato l'aumento maggiore, con il 14%, e l'elasticità rapida, che è stata la seconda funzionalità più utilizzata lo scorso anno, ha registrato l'aumento minore, con il 5%.

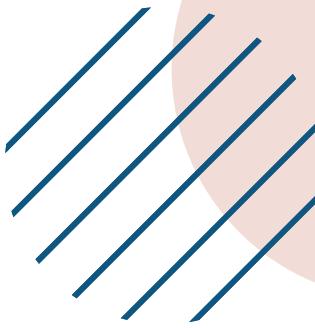


NIST	2021	2022	Variazione percentuale
<b>Ampio accesso alla rete</b>	74%	80%	8
<b>Elasticità rapida</b>	77%	81%	5
<b>Self-service on demand</b>	73%	78%	7
<b>Servizio misurato</b>	78%	83%	7
<b>Pooling di risorse</b>	73%	83%	14

  
**14%**  
di miglioramento nel  
pooling delle risorse



Dave Stanke



## SRE e DevOps

Un team tecnologico di successo contribuisce alla propria organizzazione più della distribuzione di codice, e anche più della distribuzione di codice di qualità. Garantisce inoltre che i servizi che fornisce rimangano disponibili, performanti e coerenti con le aspettative degli utenti nel tempo. L'affidabilità è una misura sfaccettata di quanto bene un team mantiene questi impegni e quest'anno abbiamo continuato le nostre esplorazioni sull'affidabilità come fattore per le prestazioni operative e di distribuzione del software.

L'SRE (Site Reliability Engineering) è un approccio influente alle operazioni che ha avuto origine in Google ed è ora praticato in molte organizzazioni. L'SRE dà la priorità all'apprendimento empirico, alla collaborazione interfunzionale, all'ampio affidamento sull'automazione e all'uso di tecniche di misurazione, inclusi gli obiettivi del livello di servizio (SLO). Altre pratiche operative moderne utilizzano metodi simili ma applicano convenzioni di denominazione diverse. Pertanto, per valutare la portata di queste pratiche nel modo più oggettivo possibile, nel testo che presentiamo agli intervistati per il nostro sondaggio utilizziamo un linguaggio descrittivo neutro. Raccogliamo anche dati sui risultati dell'ingegneria dell'affidabilità:

la misura in cui i team sono in grado di raggiungere i propri obiettivi di affidabilità. Sia gli input che gli output (pratiche SRE e risultati di affidabilità) si riflettono nel nostro modello predittivo insieme ad altre funzionalità DevOps.

### L'affidabilità è essenziale

L'adozione dell'SRE è diffusa tra i team che abbiamo intervistato: la maggior parte degli intervistati utilizza una o più delle pratiche di cui abbiamo chiesto informazioni. In tutta questa varietà di team, i dati rivelano una relazione sfumata tra affidabilità, distribuzione del software e risultati: quando l'affidabilità è scarsa, le prestazioni di distribuzione del software non sono predittive del successo dell'organizzazione. Tuttavia, con una maggiore affidabilità, iniziamo a vedere l'influenza positiva della distribuzione del software sul successo aziendale.

**Senza affidabilità, le prestazioni di distribuzione del software non prevedono il successo dell'organizzazione.**

## L'investimento in SRE produce miglioramenti in termini di affidabilità, ma solo una volta raggiunta una soglia di adozione.

Questo fenomeno è coerente con l'uso del framework SRE "budget di errore": quando un servizio è inaffidabile, gli utenti non trarranno vantaggio dall'introdurre il codice più velocemente in quel fragile contesto.

Come hanno affermato da tempo i Site Reliability Engineers, l'affidabilità è la "caratteristica" più importante di qualsiasi prodotto. La nostra ricerca supporta l'osservazione che mantenere le promesse fatte agli utenti è una condizione necessaria affinché una migliore distribuzione del software vada a vantaggio dell'organizzazione.

### Riconoscere la curva a J

Quali sfide ti aspettano sulla strada per raggiungere l'affidabilità? Nella pubblicazione O'Reilly "Enterprise Roadmap to SRE",<sup>1</sup> i collaboratori del sondaggio del team DORA James Brookbank e Steve McGhee riflettono sulle loro esperienze di implementazione di SRE in organizzazioni affermate e raccomandano di "riconoscere la curva a J del cambiamento". Descritto in precedenza nel report State of DevOps del 2018, la "curva a J" è un fenomeno in cui le trasformazioni dell'organizzazione tendono a mostrare un successo iniziale, seguito da periodi di rendimenti ridotti

o addirittura regressioni. Coloro che persistono attraverso queste sfide, tuttavia, spesso sperimentano livelli rinnovati e sostenuti di risultati elevati.

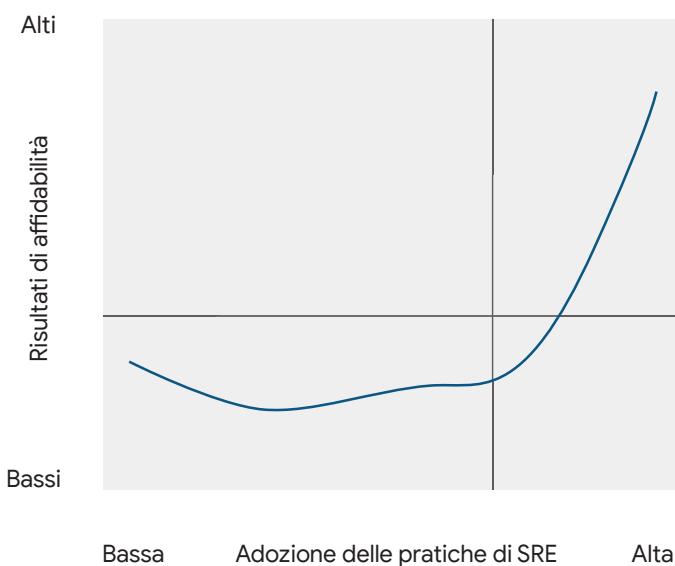
La nostra ricerca di quest'anno rivela un pattern di curva a J tra i team tecnologici che abbiamo studiato: quando i team si impegnano in meno pratiche di ingegneria dell'affidabilità, il che suggerisce che sono all'inizio del loro percorso verso l'adozione dell'SRE, queste pratiche non prevedono risultati di affidabilità migliori. Tuttavia, man mano che i team adottano più SRE, raggiungono un punto di svolta in cui l'uso dell'SRE inizia a consentire una forte previsione dell'affidabilità e, a seguire, delle prestazioni dell'organizzazione.



<sup>1</sup> <https://sre.google/resources/practices-and-processes/enterprise-roadmap-to-sre/>

## I team affidabili rendono i servizi affidabili: la cultura generativa dei team fa prevedere una migliore affidabilità.

I team che sono nelle prime fasi di un percorso verso una pratica SRE dovrebbero essere preparati alle battute d'arresto. Può essere una strada lunga poiché cultura, processo e strumenti vanno tutti riallineati a nuovi principi guida. Ma si può essere certi che, con il tempo e investimenti continui, il successo è molto probabile.



I team che persistono oltre le fasi iniziali dell'adozione dell'SRE vedono un miglioramento crescente nei risultati di affidabilità

## Investire in persone, processi e strumenti

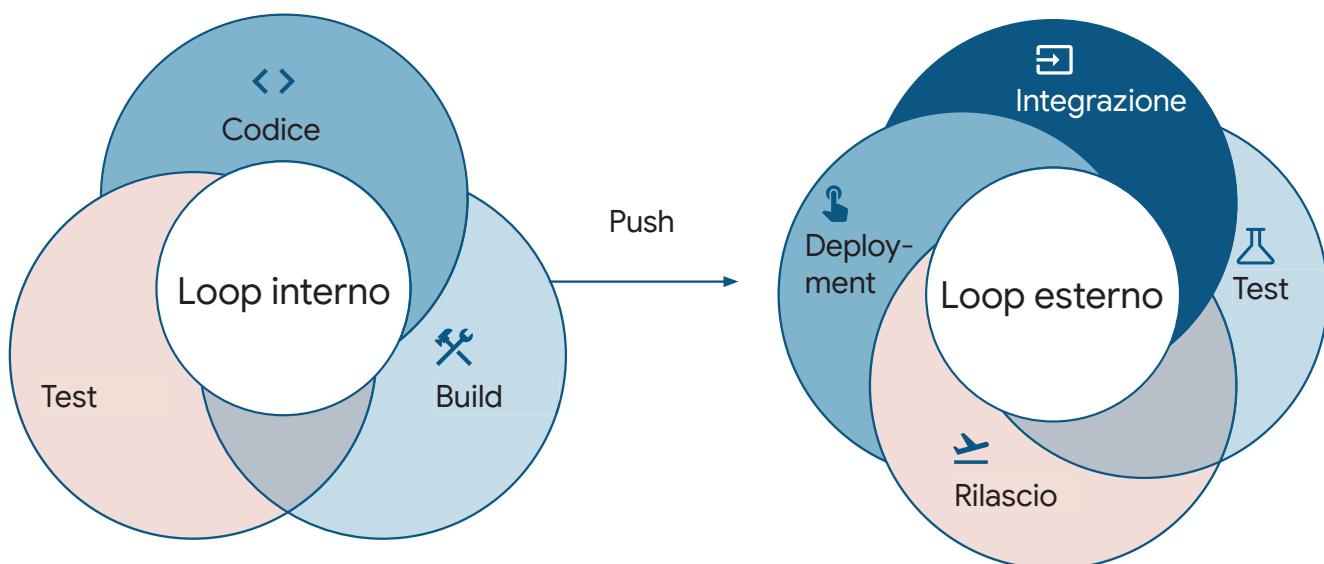
L'affidabilità è uno sforzo umano e per molti versi l'approccio SRE ne è un esempio. Uno dei principi fondamentali dell'SRE è che la percezione dell'utente, al contrario dei dati di monitoraggio interno, è la vera misura dell'affidabilità. Quindi forse non sorprende che l'affidabilità sia guidata dalle dinamiche positive del team. Abbiamo scoperto che i team con una cultura "generativa", che mostra fiducia e collaborazione, hanno maggiori probabilità di praticare l'SRE e maggiori probabilità di ottenere buoni risultati di affidabilità. Anche i team stabili, la cui appartenenza è costante nel tempo, offrono una maggiore affidabilità per i servizi rivolti agli utenti. E, come con le DevOps nel loro insieme, gli sforzi nell'ingegneria dell'affidabilità traggono vantaggio dall'aumento dell'impegno umano su processi e strumenti. Pratiche come l'uso del cloud computing e l'integrazione continua sono predittive di migliori risultati di affidabilità.

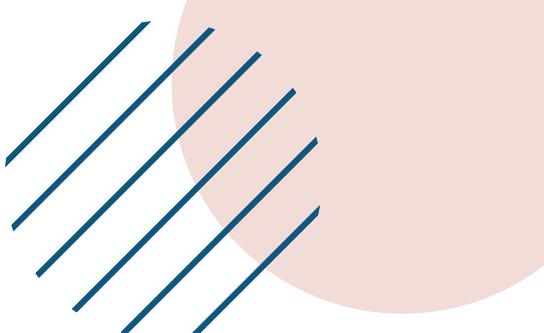


Eric Maxwell

## Funzionalità tecniche DevOps

Quest'anno abbiamo esaminato una varietà di funzionalità per comprendere i risultati che sono guidati da diverse pratiche tecniche. Abbiamo preso in considerazione due ampie fasi dello sviluppo del software: il "loop interno", che comprende attività dello sviluppatore come la codifica, il test e l'invio al controllo della versione, e il "loop esterno", che include attività come l'unione del codice, la revisione automatica del codice, esecuzione dei test, deployment e rilascio.





Gli utenti con alte prestazioni che soddisfano gli obiettivi di affidabilità hanno una **probabilità 1,4 volte maggiore di utilizzare l'integrazione continua**.

La nostra ricerca mostra che le aziende che eccellono nello sviluppo del loop interno ed esterno sono in grado di distribuire il codice più velocemente e con livelli di affidabilità più elevati. Le funzionalità che contribuiscono maggiormente alle prestazioni elevate sono il controllo della versione, l'integrazione continua, la distribuzione continua e l'architettura a basso accoppiamento.

**Le organizzazioni con prestazioni elevate che raggiungono gli obiettivi di affidabilità hanno una probabilità del:**

**33%**

maggiore di utilizzare il controllo della versione

**39%**

maggiore di applicare l'integrazione continua

**46%**

maggiore di applicare la distribuzione continua

**40%**

maggiore di avere sistemi basati su un'architettura a basso accoppiamento

Infatti, gli intervistati che fanno un uso superiore alla media di tutte le funzionalità di cui sopra hanno **prestazioni dell'organizzazione 3,8 volte superiori** rispetto a coloro che non utilizzano queste funzionalità tecniche.

## Integrazione continua

L'integrazione continua, spesso definita CI, è una parte del processo di sviluppo del loop esterno che crea automaticamente un artefatto ed esegue una serie di test automatizzati per ogni commit di codice per valutare se il codice è pronto per il deployment. Questo processo fornisce un feedback rapido e automatizzato agli sviluppatori, consentendo loro di operare con una maggiore sicurezza. L'integrazione continua è una parte fondamentale del trasferimento del codice dalla workstation di uno sviluppatore alla produzione. Come negli anni precedenti, l'integrazione continua si è dimostrata in grado di migliorare le prestazioni di distribuzione. **Le organizzazioni con prestazioni elevate che soddisfano gli obiettivi di affidabilità hanno una probabilità 1,4 volte maggiore di utilizzare l'integrazione continua rispetto alle altre.**

Quest'anno abbiamo approfondito un po' di più l'altra parte del processo di sviluppo del loop esterno: la distribuzione continua, che descriveremo in un capitolo successivo. Ma prima, diamo un'occhiata a un componente complementare all'integrazione continua: lo sviluppo basato su trunk.

## Sviluppo basato su trunk

Lo sviluppo basato su trunk è la pratica di unire continuamente il codice nel trunk ed evitare rami di funzionalità di lunga durata. Questa pratica è considerata un complemento all'integrazione continua e da anni ha dimostrato di accelerare la velocità di distribuzione del software.

A causa del cambiamento demografico di quest'anno dopo anni di esperienza nel campo, siamo in grado di vedere che l'esperienza è importante quando si implementa lo sviluppo basato su trunk. L'anno scorso il 40% degli intervistati ha dichiarato di avere più di 16 anni di esperienza nel settore e quest'anno quella categoria rappresentava solo il 13%. Continuando a convalidare il nostro tema "La distribuzione dipende", vediamo che le persone con meno esperienza complessiva hanno risultati meno positivi riguardo allo sviluppo basato su trunk e riscontrano:

- ▼ **Una diminuzione** generale nelle prestazioni di distribuzione del software
- ▲ **Un aumento** nella quantità di lavoro non pianificato
- ▲ **Un aumento** nella propensione all'errore
- ▲ **Un aumento** nel tasso di errore delle modifiche

Le persone con oltre 16 anni di esperienza che utilizzano lo sviluppo basato su trunk si rendono conto dei vantaggi della pratica e riscontrano:

- ▲ **Un aumento** generale nelle prestazioni di distribuzione del software
- ▼ **Una diminuzione** nella quantità di lavoro non pianificato
- ▼ **Una diminuzione** nella propensione all'errore
- ▼ **Una diminuzione** nel tasso di errore delle modifiche

Ciò è probabilmente dovuto alle pratiche aggiuntive necessarie per implementare correttamente lo sviluppo basato su trunk. I team che non hanno regole rigorosamente applicate per non allontanarsi mai da un trunk interrotto o che non utilizzano rami di codice ad accesso controllato e con rollback automatico in grado di interrompere il trunk andranno sicuramente incontro a problemi durante il tentativo di sviluppo sul trunk.

Tuttavia, la presenza dello sviluppo basato su trunk mostra un impatto positivo sulle prestazioni complessive dell'organizzazione.



Frank Xu

## Distribuzione continua

La distribuzione continua (CD) è una pratica di sviluppo software che:

1. Consente al team di distribuire il software in produzione o agli utenti finali in qualsiasi momento.
2. Assicura che il software sia in uno stato che consenta il deployment per tutto il suo ciclo di vita, anche quando si lavora su nuove funzionalità.
3. Stabilisce un ciclo di feedback rapido che consente al team di verificare la qualità e la possibilità di eseguire il deployment del sistema e assegna la priorità alla risoluzione dei problemi che bloccano il deployment.

Si noti che la distribuzione continua non implica necessariamente il deployment continuo, la pratica in cui viene eseguito automaticamente il deployment di ogni build del software. La distribuzione continua richiede solo che in qualsiasi momento possa essere eseguito il deployment di una build del software.

L'anno scorso, abbiamo esaminato le funzionalità tecniche DevOps che prevedono la probabilità che un team pratichi la CD e abbiamo scoperto che fattori come l'architettura a basso accoppiamento e test/integrazione continui erano tra i più forti predittori. Quest'anno, oltre a esaminare i fattori che incoraggiano l'uso della CD, abbiamo analizzato e identificato gli effetti della sola CD e le sue interazioni con altre funzionalità DevOps sui risultati dello sviluppo.

### La CD migliora le prestazioni di distribuzione del software

Analogamente ai risultati degli anni precedenti, l'uso della CD è un fattore predittivo di prestazioni di distribuzione del software più elevate, sia da sola che in combinazione con altre funzionalità DevOps. I team con un punteggio più alto sulla CD hanno maggiori probabilità di avere una maggiore frequenza di distribuzione del codice alla produzione e tempi di risposta più brevi per le modifiche e il ripristino del servizio.

**I team che combinano il controllo della versione e la distribuzione continua hanno una probabilità 2,5 volte maggiore di ottenere prestazioni di distribuzione del software elevate rispetto ai team che si concentrano solo su una delle due pratiche.**

Inoltre, gli intervistati hanno una probabilità 2,5 volte maggiore di riferire prestazioni di distribuzione del software più elevate quando il team adotta anche pratiche di controllo della versione.

## La CD può aumentare il lavoro non pianificato

I dati hanno suggerito che la distribuzione continua porta gli sviluppatori a dedicare più tempo alla rielaborazione o al lavoro non pianificato. Un'ipotesi per questa scoperta è che gli sviluppatori hanno maggiori probabilità di creare applicazioni in modo iterativo quando ci sono loop di feedback più stretti. Di conseguenza, potrebbero trovarsi alcune modifiche iterative come lavoro non pianificato sulla stessa parte del sistema. Questo lavoro può essere contemporaneamente non pianificato e causato dal feedback di un deployment precedente.

## Pratiche tecniche e CD

La nostra ricerca ha regolarmente dimostrato che un'ampia gamma di funzionalità tecniche supporta la CD. Quest'anno abbiamo esplorato cosa succede quando alcune di queste funzionalità individuali vengono utilizzate insieme alla CD. Abbiamo scoperto che lo sviluppo basato su trunk e l'architettura a basso accoppiamento, insieme alla CD, possono avere un impatto negativo sulle prestazioni di un team. Ad esempio, vediamo che i team che stanno adottando insieme architetture a basso accoppiamento e CD hanno il 43% di probabilità in più di prevedere una propensione agli errori superiore alla media (ad esempio, interruzioni del prodotto, vulnerabilità della sicurezza e significativo degrado delle prestazioni dei loro servizi) rispetto ai team che hanno adottato solo la CD. Questi effetti richiedono ulteriori indagini e indicano alcuni potenziali fastidi per i team che stanno migliorando. Questi fastidi possono essere correlati alla curva a J di trasformazione in cui i team realizzano i primi miglioramenti ma poi vacillano nello spostarsi ai passaggi successivi. L'impegno per il miglioramento è necessario per realizzare il suo pieno potenziale. Quando migliori qualsiasi funzionalità, come la CD, assicurati di osservare gli effetti sul team e le prestazioni complessive.



David Farley

## Architettura a basso accoppiamento

I sistemi a basso accoppiamento sono importanti per l'efficacia di team e organizzazioni. Questo non si applica solo ai sistemi basati su cloud o microservizi, ma ha a che fare con la capacità di un'organizzazione di apportare modifiche. La facilità con cui un'organizzazione può modificare in modo sicuro e con decisione il proprio software è un indicatore della qualità del software.

Con un'architettura a basso accoppiamento i team possono:

- Apportare modifiche su larga scala al design del loro sistema senza dover dipendere da altri team per apportare modifiche ai loro sistemi
- Ottenere un feedback più rapido attraverso test indipendenti e on demand con costi di coordinamento inferiori
- Eseguire il deployment del codice con tempi di inattività trascurabili

Nel report di quest'anno, abbiamo chiesto agli intervistati di descrivere se il software che realizzano si basa o meno su un'architettura a basso accoppiamento. Abbiamo avuto risultati interessanti, che hanno mostrato una varietà di associazioni per lo più positive tra la presenza di un'architettura a basso accoppiamento e le prestazioni dei team su più dimensioni.



## Vantaggi di un'architettura a basso accoppiamento

I team che si concentrano sulla creazione di software con architetture a basso accoppiamento sono in una posizione migliore per ottenere ottime prestazioni in termini di stabilità, affidabilità e velocità effettiva. È anche più probabile che questi team consiglino il proprio posto di lavoro a un amico o a un collega.

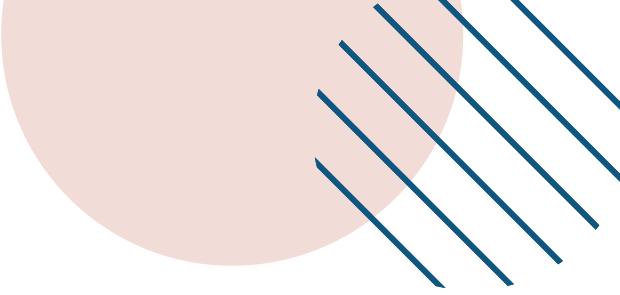
È comune vedere software che utilizza un'architettura a basso accoppiamento da parte di team che stanno eseguendo deployment nel cloud e adottando un approccio basato sull'architettura dei microservizi,

**I team che si concentrano sulla creazione di software con architetture a basso accoppiamento sono in una posizione migliore per ottenere ottime prestazioni in termini di stabilità, affidabilità e velocità effettiva.**

gestendo centinaia di servizi. Tuttavia, il basso accoppiamento è più di una semplice misura del numero di servizi in un sistema. Il deployment dei componenti in un'architettura a basso accoppiamento può essere eseguito in modo indipendente. Questa indipendenza consente ai team di sviluppare, testare ed eseguire il deployment dei propri servizi senza un costoso overhead per il coordinamento tra i team.

Nel mondo reale, il basso accoppiamento non è limitato a uno stile di architettura; fondamentalmente, si tratta della capacità di apportare modifiche in una parte del sistema senza che questa modifica influisca su altre parti. Ciò consente alle organizzazioni di suddividere il proprio lavoro, in modo che i singoli team possano fare progressi senza doversi coordinare con altri team.

In base alla nostra esperienza, i team che richiedono test di integrazione approfonditi con altri servizi per creare fiducia nel loro software prima del deployment non hanno ancora raggiunto il basso accoppiamento; per fare ciò, questi team trarrebbero vantaggio dal miglioramento delle interfacce e dell'isolamento tra i sistemi. Un modo efficace per migliorare le interfacce e l'isolamento consiste nel migliorare la "testabilità" di servizi e componenti. Se il tuo design ti consente di testare il tuo servizio in modo isolato, la sua interfaccia è, per definizione, a basso accoppiamento.



Abbiamo anche scoperto che i team coesi e stabili che utilizzano un'architettura a basso accoppiamento hanno maggiori probabilità di utilizzare pratiche di sviluppo del software che incoraggino e supportino il miglioramento continuo. Ad esempio, pratiche SRE come l'impostazione di obiettivi di affidabilità per stabilire le priorità del lavoro o l'esecuzione di revisioni periodiche per rivedere gli obiettivi di affidabilità sulla base dei risultati, supportano entrambe l'architettura a basso accoppiamento.

Le architetture a basso accoppiamento consentono inoltre all'organizzazione di aggiungere più facilmente dipendenti, poiché i team indipendenti che non hanno bisogno di coordinarsi con altri team sono più liberi di aumentare le proprie dimensioni in modo indipendente.

In breve, il basso accoppiamento dei servizi software ha un impatto che non riguarda solo l'aspetto tecnico. Influisce anche sugli aspetti socio-tecnici dello sviluppo del software. L'accoppiamento è alla base della legge di Conway: l'idea che i sistemi di progettazione di un'organizzazione rispecchino la propria struttura di comunicazione. Sistemi ad accoppiamento più basso significano organizzazioni ad accoppiamento più basso, con un approccio allo sviluppo più distribuito e scalabile.

## Risultati sorprendenti

La ricerca di quest'anno ha rivelato che l'architettura a basso accoppiamento potrebbe contribuire al burnout dei team. Questo è un risultato sorprendente che contraddice quelli degli anni precedenti. La nostra analisi mostra che i team stabili in cui le informazioni fluiscono liberamente hanno livelli più bassi di burnout. La cultura generativa e la stabilità dei team di Westrum supportano entrambe un'architettura a basso accoppiamento e riducono il burnout, quindi questo è chiaramente contraddittorio. Sono necessarie ulteriori ricerche prima di poter trarre conclusioni definitive.

Allo stesso tempo, quando i requisiti di sicurezza sono definiti e controllati da un'organizzazione di sicurezza consolidata, potrebbe essere più difficile per i team separare il proprio software dagli altri team. Ciò dimostra ulteriormente il vantaggio di trasferire i problemi di sicurezza al team che è maggiormente responsabile dell'applicazione (vedi anche: [Perché la sicurezza della catena di fornitura è importante](#)). Questa è una delle forme più sottili di accoppiamento nelle organizzazioni e, sebbene abbiano raccolto i dati sulla sicurezza, è probabile che questo valga anche per altre funzioni centralizzate. Consentire ai team di prendere le proprie decisioni sulla sicurezza e su altre funzioni spesso centralizzate è un modo per compiere progressi verso la riscossione dei vantaggi che l'utilizzo di un'architettura a basso accoppiamento può garantire alla tua organizzazione.





Daniella Villalba

## Cultura

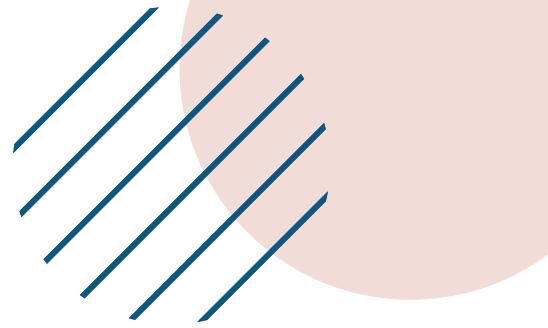
"È così che si fanno le cose qui."

È probabile che le persone abbiano pronunciato questa frase innumerevoli volte e in una vasta gamma di settori per descrivere l'approccio della loro organizzazione alle sfide e alle opportunità.

Ogni organizzazione ha la sua cultura unica e la nostra ricerca ha sempre dimostrato che la cultura è fondamentale per il successo di un'organizzazione e il benessere dei suoi dipendenti.

La cultura è inoltre un aspetto necessario delle DevOps poiché, al livello più elementare, le DevOps riguardano gli strumenti, le pratiche e **il modo in cui le persone lavorano insieme** per sviluppare e distribuire software in modo rapido, affidabile e sicuro. Comprendere i fattori che influenzano la cultura di un'organizzazione può aiutare la leadership ad affrontare le sfide legate alla cultura a testa alta. Pertanto, promuovere una cultura sana dovrebbe essere una priorità per le organizzazioni. Se non affrontate, queste sfide legate alla cultura potrebbero impedire alle pratiche DevOps di prendere piede.





Quest'anno abbiamo continuato a utilizzare la tipologia dell'organizzazione di Westrum per misurare lo stato di salute della cultura di un'organizzazione. Inoltre, abbiamo ampliato la nostra comprensione della cultura misurando il tasso di abbandono dei team, gli accordi di lavoro flessibili, il coinvolgimento dell'organizzazione percepito e il burnout.

I dati della ricerca di quest'anno supportano i risultati precedenti secondo cui le **prestazioni dell'organizzazione** sono influenzate dal tipo di cultura esistente all'interno di un'organizzazione. Nello specifico, una **cultura generativa** è associata a livelli più elevati di prestazioni dell'organizzazione rispetto alle organizzazioni caratterizzate da una cultura burocratica o patologica. I dipendenti delle organizzazioni con una cultura

generativa hanno maggiori probabilità di appartenere a team stabili, produrre documentazione di qualità superiore e trascorrere la maggior parte del loro tempo impegnati in un lavoro significativo.

## Tasso di abbandono del team

Abbiamo esaminato il tasso di abbandono del team e abbiamo scoperto che l'esistenza di **team stabili**, ovvero i team la cui composizione non era cambiata molto negli ultimi 12 mesi, è più probabile all'interno di organizzazioni ad alte prestazioni. L'abbandono costante può influire sulla produttività e sul morale poiché i nuovi membri del team hanno bisogno di tempo per l'onboarding. E coloro che restano potrebbero doversi adattare a cambiamenti dei

## La cultura dell'organizzazione di Westrum

Patologica Orientata al potere	Burocratica Orientata alle regole	Generativa Orientata alle prestazioni
Cooperazione scarsa	Cooperazione modesta	Cooperazione elevata
Chi porta messaggi viene rifiutato	Chi porta messaggi viene ignorato	Chi porta messaggi viene formato
Le responsabilità vengono evitate	Responsabilità limitate	Le responsabilità vengono condivise
La collaborazione viene scoraggiata	La collaborazione viene tollerata	La collaborazione viene incoraggiata
Un fallimento richiede un capro espia-torio	Un fallimento richiede giustizia	Un fallimento richiede un approfondi-mento
Le novità vengono schiacciate	Le novità creano problemi	Le novità vengono implementate



carichi di lavoro e delle dinamiche del team. Inoltre, la nostra ricerca ha dimostrato che i team stabili hanno maggiori probabilità di produrre documentazione di qualità rispetto ai team che hanno sperimentato maggiormente l'abbandono. Un team che si trova ad affrontare costanti cambiamenti può avere difficoltà a tenere il passo con le pratiche che portano a una documentazione di qualità.

**Le organizzazioni con prestazioni elevate hanno maggiori probabilità di avere accordi di lavoro flessibili.**

## Condizioni lavorative flessibili

Dato il passaggio a modalità di lavoro flessibili che molte organizzazioni hanno adottato dallo scoppio della pandemia di COVID-19, abbiamo esaminato se dare dipendenti la libertà di scegliere tra lavoro in remoto,

in presenza o ibrido fosse un tratto associato a prestazioni dell'organizzazione elevate. I risultati hanno mostrato che le organizzazioni con livelli più elevati di **flessibilità dei dipendenti** hanno prestazioni dell'organizzazione più elevate rispetto alle organizzazioni con accordi di lavoro più rigidi. Questi risultati forniscono la prova che dare ai dipendenti la libertà di modificare i propri accordi di lavoro in base alle esigenze comporta vantaggi tangibili e diretti per un'organizzazione.

## Burnout

Il burnout è una sensazione di ansia, apatia e cinismo che riguarda il lavoro. Quando le persone sperimentano il burnout, non sono solo demotivate ed esauste, ma hanno anche maggiori probabilità di avere una minore soddisfazione sul lavoro, che può aumentare il turnover dei dipendenti. Il burnout è stato collegato a un'ampia gamma di risultati negativi riguardanti la salute psicofisica, ad esempio un aumento del rischio di depressione e ansia, malattie cardiache e pensieri suicidi<sup>1</sup>.

L'anno scorso abbiamo misurato il burnout nel contesto della pandemia di COVID-19 e abbiamo scoperto che una cultura generativa era associata a tassi più bassi di burnout dei dipendenti.

<sup>1</sup> Maslach C, Leiter MP. Understanding the burnout experience: recent research and its implications for psychiatry. World Psychiatry. 2016 Jun;15(2):103-11. doi: 10.1002/wps.20311. PMID: 27265691; PMCID: PMC4911781.

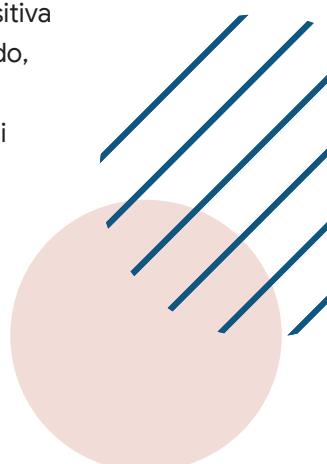
I modelli di lavoro flessibili sono associati alla diminuzione del burnout dei dipendenti e all'aumento della probabilità dei dipendenti di raccomandare il proprio team come un buon posto di lavoro.

Quest'anno abbiamo replicato questi risultati e approfondito la nostra comprensione del burnout, dimostrando che anche team stabili e accordi di lavoro flessibili sono associati a meno burnout. Inoltre, quest'anno abbiamo misurato il Net promoter score (NPS) del team, che indica se le persone consiglierebbero il proprio team a un amico o a un collega. Abbiamo scoperto che l'NPS del team è associato al coinvolgimento percepito della leadership. E rispecchiando i risultati del burnout, abbiamo scoperto che una cultura generativa, un team stabile e un accordo di lavoro flessibile sono associati a persone che sono più propense a raccomandare il proprio team ad altri.

## Come i dipendenti percepiscono la loro organizzazione

Infine, abbiamo esaminato il coinvolgimento percepito della leadership chiedendo alle persone di prevedere quanto supporto si aspettavano che il loro team ricevesse nei prossimi 12 mesi. I risultati hanno mostrato che un maggior coinvolgimento percepito della leadership (ad esempio maggiore sostegno finanziario, maggiore allocazione di risorse, sponsorizzazioni) era associato a organizzazioni con prestazioni elevate.

Abbiamo anche chiesto alle persone di prevedere la probabilità che si verifichi una violazione della sicurezza o un'interruzione completa nei prossimi 12 mesi. I risultati hanno mostrato che le persone che lavorano in organizzazioni con prestazioni elevate erano meno propense ad aspettarsi che si verificasse un errore grave: avevano una visione più positiva della loro organizzazione. Allo stesso modo, abbiamo scoperto che le persone che lavorano in organizzazioni con prestazioni del software e di distribuzione elevate erano **meno propense** a ritenere che le loro pratiche correnti dovessero essere modificate per migliorare i risultati aziendali.



## Qualche parola sulla rappresentanza

I nostri risultati hanno mostrato che i dipendenti di gruppi sottorappresentati avevano maggiori probabilità di dedicare più tempo a lavori non pianificati indipendentemente dal fatto che appartenessero a organizzazioni con prestazioni basse o elevate. Abbiamo anche scoperto che i dipendenti di gruppi sottorappresentati hanno riportato livelli di burnout più elevati rispetto ai dipendenti che non appartengono a gruppi sottorappresentati. I responsabili dei team devono essere consapevoli del rischio di squilibrio del carico di lavoro e garantire che il lavoro sia distribuito equamente tra i membri del team.

Nel loro insieme, questi risultati sottolineano l'importanza di creare un ambiente sano e inclusivo per i dipendenti sia a livello dell'organizzazione che di team.

Pur continuando a sottolineare l'importanza della cultura, riconosciamo che modificare o anche migliorare la cultura di un'organizzazione non è un compito facile. Consigliamo alle organizzazioni di cercare prima di tutto di comprendere le esperienze dei propri dipendenti e successivamente di investire risorse nell'affrontare i problemi legati alla cultura come parte degli sforzi per la trasformazione delle DevOps.



# 04

## Perché la sicurezza della catena di fornitura è importante

Nel novembre 2020, relativamente pochi professionisti della tecnologia sospettavano che si stesse preparando una crisi della sicurezza della catena di fornitura del software. La [Open Source Security Foundation](#), un successore di sforzi passati, era stata fondata per concentrarsi sulla sicurezza del software open source, e mentre c'erano alcuni [latipositi](#) per affrontare questo problema, l'argomento non era sulla prima pagina dei

principali quotidiani. Un grave attacco, [SolarWinds](#), ha cambiato tutto. Quando gli utenti malintenzionati riescono a penetrare silenziosamente in migliaia di grandi aziende e reti governative grazie ad aggiornamenti software trojan, i tempi cambiano rapidamente.

Oggi, la sicurezza della catena di fornitura del software è ampiamente riconosciuta come un argomento da discutere urgentemente, se non durante le cene in famiglia, sicuramente durante le riunioni aziendali.



John Speed Meyers Todd Kulesza



Esistono numerose iniziative e gran parte dell'industria del software si è impegnata a riformare le proprie pratiche di sicurezza della catena di fornitura del software e migliorare la sicurezza delle risorse comuni open source.

In questo capitolo, ci concentriamo su due iniziative: Supply Chain Levels for Software Artifacts ([SLSA](#), pronunciato "salsa"), e il NIST Secure Software Development Framework ([SSDF](#)). Ciascuno offre una gamma di misure difensive per assicurarsi che utenti malintenzionati non possano manomettere i processi di produzione del software e superare le misure di difesa della rete tramite aggiornamenti software dannosi.

Ma quanto sono diffuse le pratiche di sicurezza della catena di fornitura del software associate a SLSA e SSDF? Quali pratiche hanno bisogno di un aiuto che ne incentivi l'adozione e quali sono già ampiamente utilizzate? Fino a oggi non c'erano risposte sistematiche a queste domande. Interrogando centinaia di professionisti del software sul loro utilizzo delle pratiche associate alla sicurezza della catena di fornitura, forniamo alcune risposte iniziali. In particolare, emergono quattro risultati principali:

**01 L'adozione è già iniziata:** le pratiche di sicurezza della catena di fornitura del software incorporate in SLSA e SSDF vedono già un'adozione modesta, ma c'è ampio spazio per altro.

**02 Le culture più sane hanno un vantaggio:** la cultura dell'organizzazione è uno dei motori principali delle pratiche di sicurezza dello sviluppo software, con una maggiore fiducia, le culture "senza attribuzione di colpe" hanno maggiori probabilità di stabilire pratiche SLSA e SSDF rispetto alle culture dell'organizzazione con fiducia ridotta.

**03 C'è un punto chiave per l'integrazione:** l'adozione degli aspetti tecnici della sicurezza della catena di fornitura del software sembra dipendere dall'uso di un approccio di CI/CD, che spesso fornisce la piattaforma di integrazione per molte pratiche di sicurezza della catena di fornitura.

**04 Offre vantaggi inaspettati:** oltre a una riduzione dei rischi per la sicurezza, migliori pratiche di sicurezza comportano ulteriori vantaggi, come la riduzione del burnout.

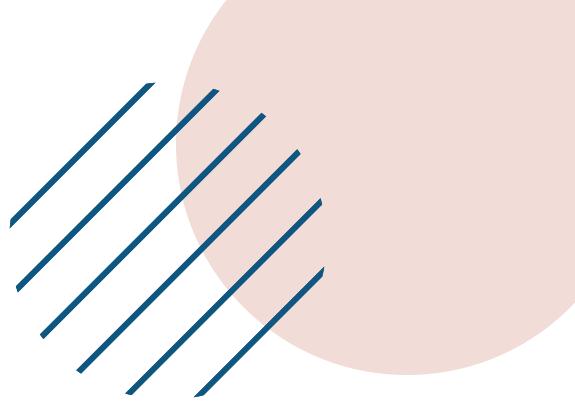
## Cosa fanno oggi le aziende per evitare le vulnerabilità di sicurezza

Per comprendere meglio cosa stanno facendo le organizzazioni oggi per identificare e risolvere le vulnerabilità di sicurezza nel software che creano, abbiamo aggiunto oltre due dozzine di domande al sondaggio di quest'anno. Queste domande rientravano sostanzialmente in due categorie:

- Domande che chiedevano agli intervistati se fossero d'accordo o in disaccordo con un'affermazione (ad esempio, "La mia organizzazione dispone di un metodo efficace per affrontare le minacce alla sicurezza" o "Ho accesso agli strumenti necessari per eseguire i test di sicurezza").
- Domande che chiedevano agli intervistati quanto fossero consolidate o meno le pratiche di sicurezza nella loro organizzazione (ad esempio, "Le build sono definite tramite script di build e nient'altro" o "Le release in produzione sono create utilizzando un sistema CI/CD centralizzato, ma sulla workstation di uno sviluppatore"). Abbiamo utilizzato la scala "stabilito/non stabilito" perché i primi test hanno rilevato che gli intervistati avevano la tendenza a concordare con alcune domande relative alla sicurezza. Le domande su SSDF, tuttavia, sono state formulate in modo più naturale con la scala di risposta d'accordo/in disaccordo.

Il **framework SLSA**, alla versione 0.1 al momento della stesura di questo documento, descrive una serie di pratiche di integrità della catena di fornitura del software associate ai "livelli" di SLSA, con livelli più alti corrispondenti a livelli più elevati di garanzia della sicurezza della catena di fornitura del software. Abbiamo chiesto agli intervistati informazioni su molte delle pratiche specifiche associate a SLSA. In particolare, il sondaggio chiedeva: "Quanto sono consolidate le seguenti pratiche per l'applicazione o il servizio principale su cui lavori ?" Nella tabella 1 sono elencate le definizioni delle pratiche relative a SLSA trattate nel sondaggio.

L'**SSDF**, attualmente alla versione 1.1, è incentrato sulle pratiche per aiutare le organizzazioni a fornire software con meno vulnerabilità e per ridurre al minimo il potenziale impatto delle vulnerabilità rimanenti. Invece dei "livelli" di SLSA, le pratiche SSDF sono raggruppate in quattro categorie: preparazione dell'organizzazione, protezione del software in fase di sviluppo, produzione di software ben protetto e risposta efficace alle vulnerabilità scoperte. Il sondaggio ha chiesto agli intervistati quanto fossero d'accordo (o in disaccordo) con le affermazioni che descrivono diverse pratiche SSDF; queste domande sono riassunte nella Tabella 2.



Pratica SLSA	Definizione del sondaggio
<b>Sistema CI/CD centralizzato</b>	Le release in produzione vengono create utilizzando un sistema CI/CD centralizzato, ma sulla workstation di uno sviluppatore
<b>Conservazione della cronologia</b>	Le revisioni e la relativa cronologia delle modifiche vengono conservate a tempo indeterminato
<b>Script di build</b>	Le build sono completamente definite tramite lo script di build e nient'altro
<b>Isolamento</b>	Le build sono isolate; non possono interferire con build simultanee o successive
<b>File di testo di build</b>	Le definizioni e le configurazioni di build sono definite in file di testo archiviati in un sistema di controllo della versione
<b>Metadati dei parametri</b>	I metadati di build (ad esempio dipendenze, processo di build, ambiente di build) relativi a un artefatto includono tutti i parametri di build
<b>Metadati delle dipendenze</b>	I metadati di build (ad es. dipendenze, processo di build, ambiente di build) relativi a un artefatto documentano tutte le dipendenze
<b>Generazione dei metadati</b>	I metadati di build (ad esempio dipendenze, processo di build, ambiente di build) vengono generati dal servizio di build o da un generatore di metadati di build che legge il servizio di build
<b>Prevenzione degli input</b>	Quando si eseguono le build, ai passi di build viene impedito di caricare dinamicamente qualsiasi input di build (ovvero tutte le origini e le dipendenze richieste vengono recuperate in anticipo)
<b>Nessuna modifica da parte degli utenti</b>	I metadati di build (ad es. dipendenze, processo di build, ambiente di build) relativi a un artefatto non possono essere modificati dagli utenti dei servizi di build
<b>Metadati disponibili</b>	I metadati di build (ad es. dipendenze, processo di build, ambiente di build) sono disponibili per le persone che ne hanno bisogno (ad es. tramite un database centrale) e vengono forniti in un formato accettato
<b>Revisione di due persone</b>	Ogni modifica nella cronologia di una revisione deve essere rivista e approvata individualmente da due persone fidate prima dell'invio
<b>Metadati firmati</b>	I metadati di build (ad es. dipendenze, processo di build, ambiente di build) relativi a come un artefatto è stato prodotto vengono firmati dal mio servizio di build

**Tabella 1. Domande del sondaggio relative a SLSA**

Nota: gli intervistati avevano cinque possibili risposte a ciascuna domanda: per niente implementata, poco implementata, moderatamente implementata, molto implementata e completamente implementata.

Pratica SSDF	Definizione del sondaggio
<b>Verifiche della sicurezza</b>	Viene condotta una verifica della sicurezza per tutte le principali funzionalità delle applicazioni su cui lavoro
<b>Analisi/test continuo del codice</b>	Ci impegniamo continuamente in analisi e test automatici o manuali del codice per tutte le versioni supportate, al fine di identificare o confermare la presenza di vulnerabilità precedentemente non rilevate
<b>Test di sicurezza precoci</b>	I test di sicurezza vengono eseguiti all'inizio del processo di sviluppo del software, da me o da un altro team
<b>Affrontare efficacemente le minacce</b>	La mia organizzazione dispone di un metodo efficace per affrontare le minacce alla sicurezza
<b>Integrazione con il team di sviluppo</b>	I ruoli di sicurezza sono integrati nel nostro team di sviluppo software
<b>Requisiti della documentazione</b>	La nostra organizzazione dispone di processi per identificare e documentare tutti i requisiti di sicurezza per il software che la nostra organizzazione sviluppa o acquisisce (compresi quelli di terze parti e open source)
<b>Revisione periodica dei requisiti</b>	I requisiti di sicurezza vengono rivisti a intervalli regolari (annualmente o prima se necessario)
<b>Generazione dei metadati</b>	I metadati di build (ad esempio dipendenze, processo di build, ambiente di build) vengono generati dal servizio di build o da un generatore di metadati di build che legge il servizio di build
<b>Integrazione con il ciclo di sviluppo</b>	Nella mia azienda il protocollo di sicurezza del software è perfettamente integrato nel nostro processo di sviluppo
<b>Processo standard per tutti i progetti</b>	Nella mia azienda, abbiamo un processo standardizzato per affrontare la sicurezza del software in tutti i progetti
<b>Monitoraggio dei report di sicurezza</b>	Ci impegniamo costantemente per monitorare le informazioni provenienti da fonti pubbliche relative a possibili vulnerabilità nel software che utilizziamo e nei suoi componenti di terze parti
<b>Disponibilità degli strumenti necessari</b>	Ho accesso agli strumenti necessari per eseguire i test di sicurezza

**Tabella 2. Domande del sondaggio relative a SSDF**

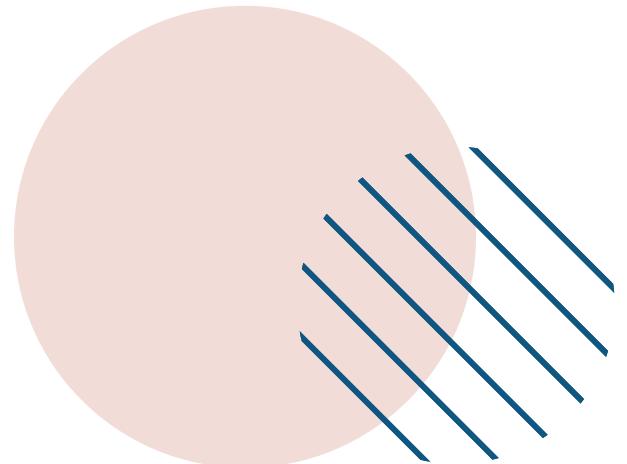
Nota: gli intervistati avevano sette possibili risposte per ciascuna domanda: totalmente in disaccordo, in disaccordo, abbastanza in disaccordo, né d'accordo né in disaccordo, abbastanza d'accordo, d'accordo e totalmente d'accordo.

Nel complesso, abbiamo riscontrato un'adozione relativamente ampia delle pratiche del settore emergente, anche se c'è molto spazio per una maggiore implementazione. Ad esempio, mentre il 66% degli intervistati è d'accordo con l'affermazione "Nella mia azienda, il protocollo di sicurezza del software è perfettamente integrato nel nostro processo di sviluppo", solo il 18% è totalmente d'accordo. Le figure 1 e 2 riassumono le risposte dei partecipanti alle nostre domande sulla sicurezza.

Abbiamo scoperto che l'utilizzo di sistemi di integrazione continua/distribuzione continua (CI/CD) per i rilasci in produzione è la pratica più comunemente implementata, con il 63% degli intervistati che ha affermato che fosse "molto" o "completamente" implementata. Il fatto che l'approccio CI/CD sia in cima a questo elenco è in linea con la precedente ricerca sulla sicurezza, che ha rilevato che [la maggior parte delle organizzazioni implementa la scansione di sicurezza a livello di applicazione come parte del proprio processo CI/CD](#). Inoltre, una serie separata di sondaggi qualitativi incentrati sulla sicurezza ha suggerito che la maggior parte degli sviluppatori non potesse eseguire questi strumenti in locale durante lo sviluppo. Allo stesso modo, il framework SLSA si basa sui sistemi CI come punto di integrazione centrale per la sicurezza della catena di fornitura. La nostra analisi del modello, descritta nella sezione successiva, ha rilevato che la presenza di CI in

un'organizzazione è un preditore della maturità delle sue pratiche di sicurezza. Pertanto, riteniamo che senza questa parte critica dell'infrastruttura, sia molto difficile per un'organizzazione garantire che un set coerente di scanner, linter e test venga eseguito sugli artefatti software che crea.

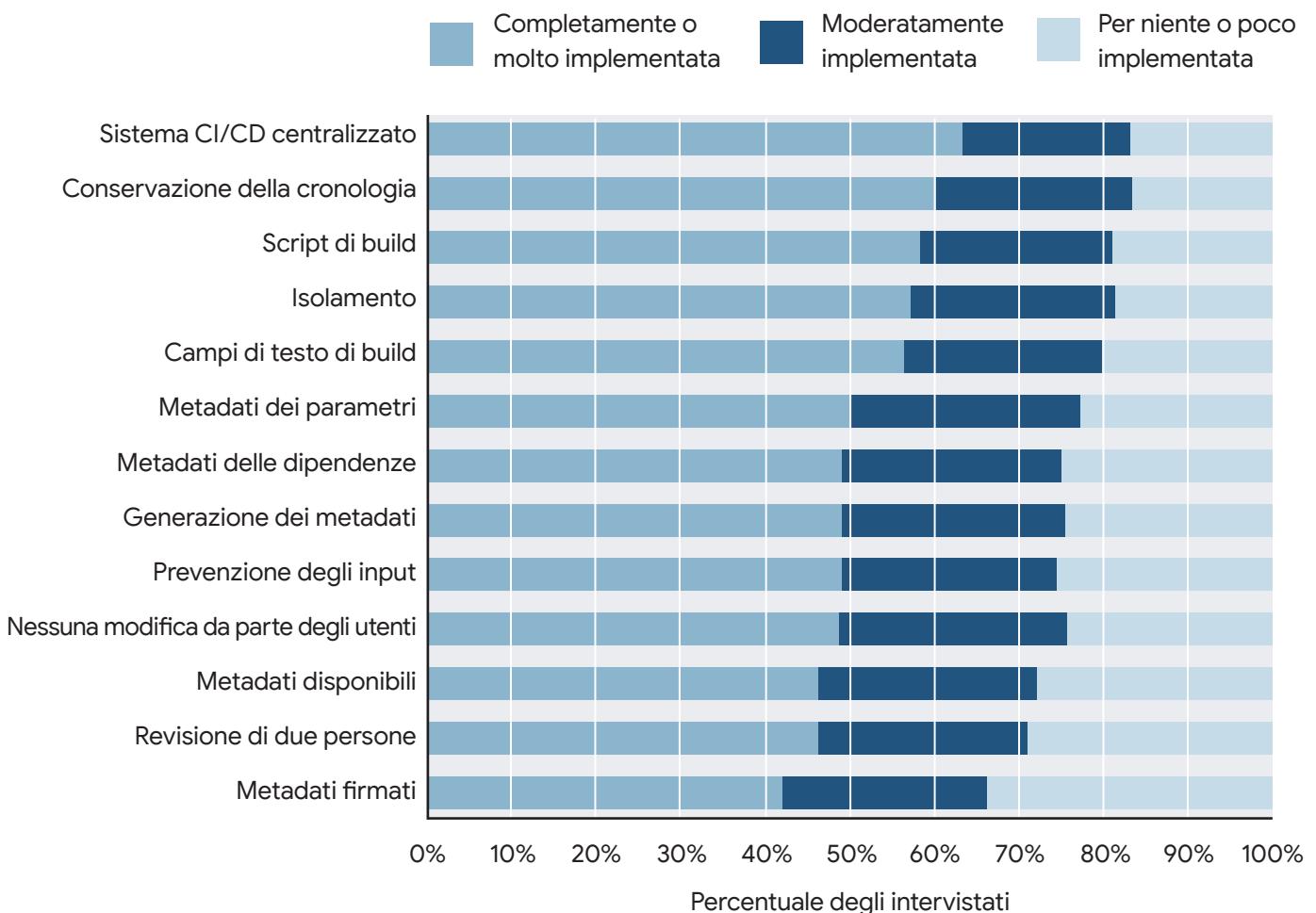
Oltre ai sistemi CI/CD, altre pratiche comunemente stabilite includono la conservazione a tempo indeterminato della cronologia del codice (60%), le build definite esclusivamente tramite script (58%), il mantenimento delle build isolate l'una dall'altra (57%) e l'archiviazione delle definizioni di build nel controllo del codice sorgente (56%). All'estremità inferiore, le due pratiche meno comunemente stabilite richiedevano a due o più revisori di approvare ogni modifica del codice (45%) e di firmare i metadati di build per prevenire/rilevare le manomissioni (41%).



Oltre alle domande sulle pratiche consolidate, abbiamo anche chiesto ai partecipanti se fossero d'accordo o in disaccordo con una serie di affermazioni sulla sicurezza nella loro organizzazione. La dichiarazione con il più alto livello di accordo è stata: "Stiamo continuando a impegnarci per monitorare le informazioni provenienti da fonti pubbliche riguardanti possibili vulnerabilità nel software che utilizziamo e nei suoi componenti di terze parti", con l'81% degli intervistati che si è dichiarato d'accordo. Al contrario, l'affermazione con la percentuale più bassa di consenso riguarda gli impatti negativi delle pratiche di sicurezza sullo sviluppo del software: il 56% degli intervistati concorda sul fatto che "i processi di sicurezza del software che esistono nella mia azienda rallentano il processo di sviluppo delle applicazioni su cui lavoro".

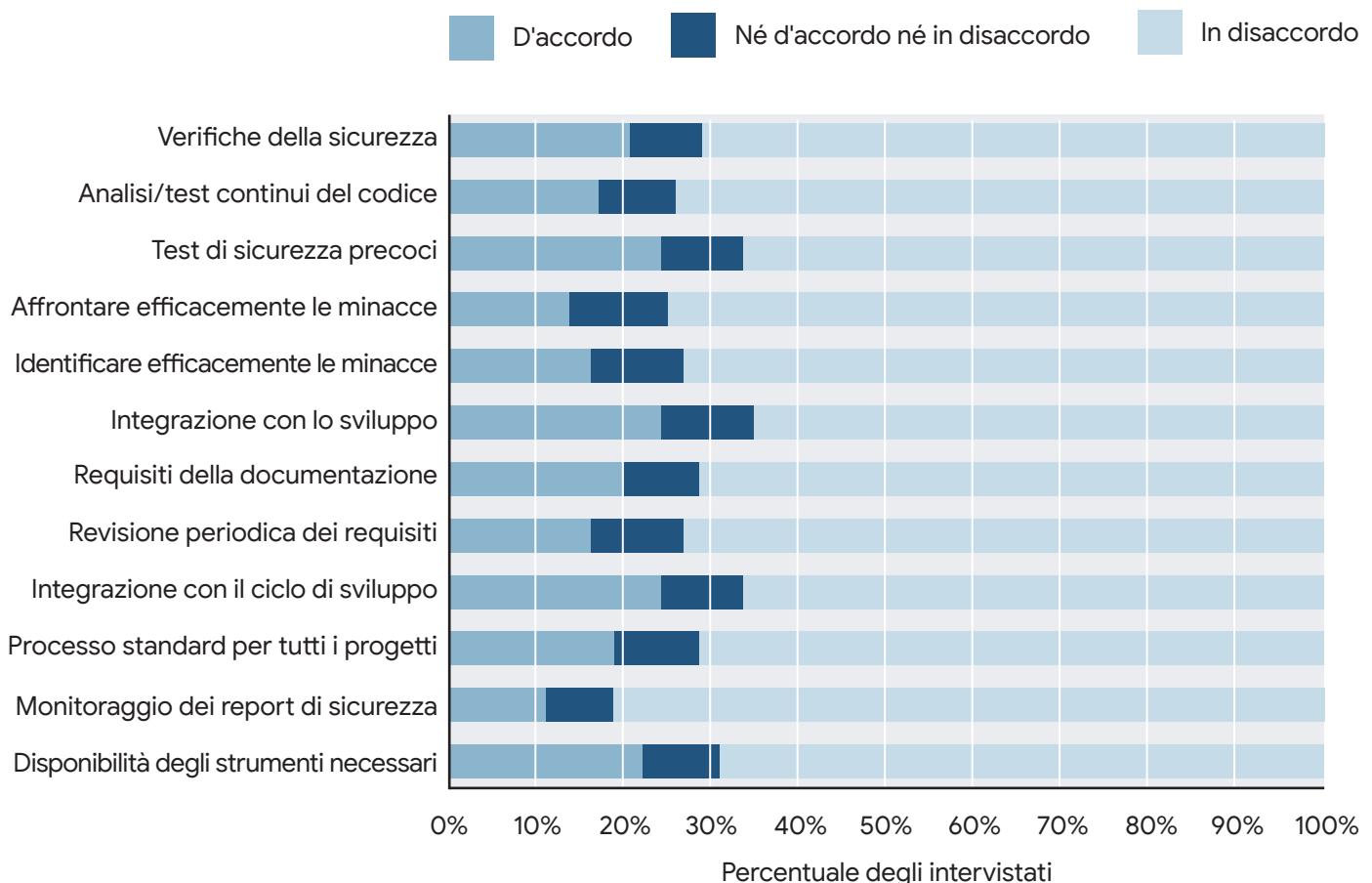
Sebbene sia incoraggiante che questa affermazione abbia avuto il livello più basso di consenso da parte degli intervistati, il fatto che la maggioranza degli intervistati abbia affermato che gli attuali processi di sicurezza rallentano lo sviluppo suggerisce un ampio margine di miglioramento negli strumenti e negli approcci per la sicurezza. Anche la nostra analisi del modello supporta questa interpretazione, mostrando effetti misti (sebbene minori) sulle prestazioni di distribuzione del software.





**Figura 1. Implementazione delle pratiche SLSA**

Risposte al sondaggio sull'implementazione di pratiche SLSA. La maggioranza degli intervistati ha indicato un certo grado di implementazione di tutte queste pratiche, ma relativamente pochi hanno affermato che fossero già "completamente" implementate.



**Figura 2. Implementazione delle pratiche SSDF**

Risposte al sondaggio sull'implementazione di pratiche SSDF. Analogamente al framework SLSA, la maggioranza degli intervistati ha convenuto che la propria organizzazione ha seguito tutte queste pratiche.

## Cosa aiuta le aziende a seguire buone pratiche di sicurezza?

La sicurezza delle applicazioni è solo un aspetto dello sviluppo del software, e quindi una delle tante richieste in competizione che richiedono il tempo e l'attenzione degli sviluppatori. Gli approcci ad alto attrito alla sicurezza possono essere frustranti per gli sviluppatori e complessivamente inefficaci, poiché le persone cercano di evitare i fastidi. Ad esempio, una serie di sondaggi di ricerca rivolti a ingegneri informatici professionisti ha rilevato che i loro punti di contatto con i team di sicurezza erano limitati all'inizio o alla fine di un progetto e che l'interazione tra i team può essere difficile. Nelle parole di un partecipante, "Abbiamo un team per la sicurezza delle applicazioni, ma non ho mai fatto rivedere il mio codice da loro... Faccio come la maggior parte degli ingegneri, di solito li evito".

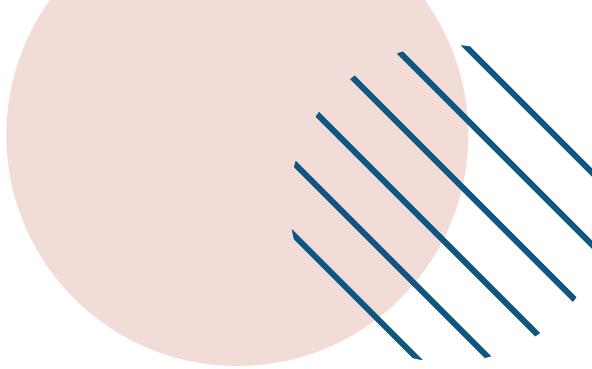
Un approccio per migliorare la sicurezza del software è ridurre gli ostacoli al rispetto delle pratiche di sicurezza. Gli sviluppatori con cui abbiamo parlato volevano fare la cosa giusta e spesso hanno discusso della frustrazione per il fatto che le funzionalità di distribuzione o le correzioni avessero costantemente la priorità rispetto ai potenziali problemi di sicurezza. Ad esempio, una delle persone intervistate per un altro sondaggio relativo alla sicurezza ha descritto la sua più grande sfida alla sicurezza come "renderla una priorità prima di tutto. Non è attraente, non vende più prodotti, [e] non è un problema finché non lo diventa".

I dati del nostro sondaggio suggeriscono che ci sono diversi fattori che rendono più facile per gli sviluppatori "fare la cosa sicura".

Il fattore più importante che abbiamo riscontrato non era affatto tecnico, ma piuttosto culturale: **le organizzazioni più vicine al gruppo culturale "generativo" Westrum erano molto più propense a dichiarare di avere pratiche di sicurezza ampiamente implementate**, come definito dal framework SLSA<sup>1</sup>. Gli aspetti delle culture generative includono l'essere fortemente collaborativi, la condivisione di rischi e responsabilità e l'apprendimento dagli errori del passato. Ipotizziamo che questi tratti si manifestino in pratiche di sicurezza più sane in numerosi modi, come incoraggiare gli ingegneri del software a essere più proattivi sulla sicurezza della catena di fornitura, premiare le persone per i loro sforzi in materia di sicurezza indipendentemente dal loro ruolo lavorativo o ridurre i rischi percepiti del segnalare potenziali problemi di sicurezza.

Dal punto di vista tecnologico, tre dei fattori più importanti alla base della sicurezza riguardano l'infrastruttura. Intuitivamente questo ha senso: se la tua infrastruttura rende più facili da condurre attività come l'analisi delle vulnerabilità o le revisioni manuali del codice, allora è più probabile che i tuoi ingegneri le utilizzino. Nello specifico, abbiamo scoperto che **la presenza di sistemi per il controllo del codice sorgente, l'integrazione continua e**

<sup>1</sup> È interessante notare che questi stessi intervistati non erano più propensi a concordare con le domande relative al framework NIST SSDF. Anche se SLSA e SSDF trattano diversi aspetti della sicurezza dello sviluppo di applicazioni, ci aspettavamo di trovare una sovrapposizione tra queste serie di domande. Come accennato in precedenza, è possibile che la scala delle risposte per il framework SSDF fosse sbilanciata verso le risposte in "accordo", il che spiegherebbe questa differenza.



## la distribuzione continua era sempre collegata anche con una maggiore implementazione delle pratiche

**SLSA.** Una parte fondamentale di questo discorso probabilmente si ha quando i problemi di sicurezza ottengono l'attenzione degli sviluppatori, che un altro sondaggio ha rilevato accadere principalmente durante la CI. In genere la CI precede direttamente le revisioni del codice ed è il momento in cui vengono eseguiti l'analisi delle vulnerabilità e altri strumenti di analisi del codice, in quanto garantisce che tutti i commit del codice siano soggetti agli stessi requisiti di sicurezza. La mancanza di un sistema di build centralizzato rende un'analisi così omogenea molto più impegnativa e la mancanza di controllo del codice sorgente, a sua volta, rende difficile avere prima di tutto un sistema di build centralizzato.

L'analisi della sicurezza come parte del sistema di CI/CD, tuttavia, potrebbe non essere abbastanza precoce per gli ingegneri informatici. In una serie di sondaggi relativi alla sicurezza, gli sviluppatori di applicazioni hanno uniformemente affermato che l'analisi della sicurezza sulla loro workstation di sviluppo avrebbe aiutato a risparmiare tempo e fatica. Due situazioni sono state comunemente citate: 1) voler sapere in anticipo se si stessero basando su una dipendenza con vulnerabilità note, in modo da poter rivalutare l'utilizzo di quella dipendenza prima di basarci sopra il loro codice, e 2) evitare lunghi tempi di attesa della CI, a volte misurati in ore, solo per verificare se le modifiche apportate avessero risolto un problema di sicurezza. In entrambi i casi, gli ingegneri informatici hanno affermato che mentre era necessario

una "rete di sicurezza" per la CI, la possibilità di eseguire gli stessi strumenti di sicurezza in locale li avrebbe aiutati a lavorare in modo più rapido ed efficiente.

I fattori culturali e tecnologici discussi sopra sono stati i fattori principali per la sicurezza, ma non gli unici. Altri fattori degni di nota:

- Flessibilità negli accordi di lavoro (ad esempio, l'organizzazione supporta il lavoro da casa?)
- Utilizzo del cloud (pubblico o privato)
- Lavorare su un servizio o un'applicazione "cloud-native"
- Sentire che l'azienda apprezza e investe nel tuo team
- Ricambio basso nel team
- Dimensioni dell'organizzazione, con le organizzazioni più grandi che riferiscono punteggi di sicurezza più elevati

Questi fattori, tuttavia, sembrano per lo più correlati alla cultura generativa di Westrum (ad esempio, accordi di lavoro flessibili, sentirsi apprezzati dalla propria organizzazione o avere un basso ricambio nel team) o all'utilizzo di CI/CD (ad esempio, lavorare su un'applicazione cloud-native o in una grande organizzazione). Questi dati ci portano a ritener che la cultura dell'organizzazione e i moderni processi di sviluppo (come l'integrazione continua) siano i maggiori driver della sicurezza dello sviluppo di applicazioni per un'organizzazione e il miglior punto di partenza per le organizzazioni che desiderano aumentare il proprio livello di sicurezza.

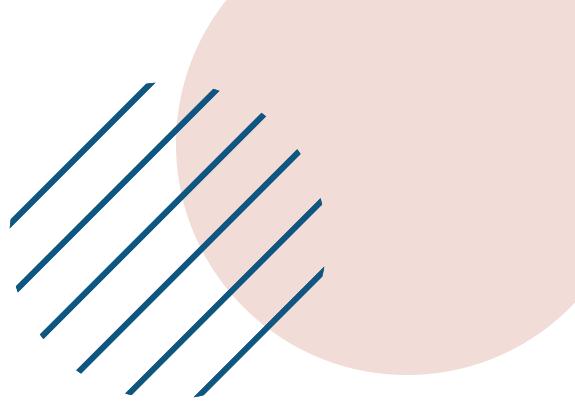
## A quali risultati portano le buone pratiche di sicurezza?

Man mano che le organizzazioni migliorano le loro pratiche di sicurezza relative allo sviluppo del software, quali vantaggi potrebbero aspettarsi di riscontrare? I dati del nostro sondaggio confermano che i partecipanti prevedono una **minore possibilità di violazioni della sicurezza, interruzioni del servizio e degrado delle prestazioni man mano che le aziende aumentano l'implementazione di pratiche di sicurezza della catena di fornitura**. Allo stesso modo, abbiamo condotto un'altra ricerca nella prima metà del 2022, scoprendo che l'esecuzione di strumenti come l'analisi delle vulnerabilità durante la CI aumenta significativamente la probabilità di identificare vulnerabilità nelle dipendenze software: gli intervistati che utilizzano questi strumenti hanno quasi il doppio delle probabilità di identificare una vulnerabilità della sicurezza nel proprio codice o in una delle sue dipendenze. In breve: Le pratiche SLSA e SSDF sembrano funzionare come previsto. Non affermiamo

che possano eliminare le minacce alla sicurezza, ma i nostri riscontri suggeriscono che sono in grado di ridurre il rischio per la sicurezza di un'organizzazione.

Le pratiche di sicurezza possono anche avere un impatto positivo sui risultati basati sulle prestazioni, ma c'è una svolta: la CI gioca un ruolo fondamentale. Quando la CI non è implementata, le pratiche di sicurezza sembrano non avere nessun effetto sulle prestazioni di distribuzione del software. Ma quando l'integrazione continua viene implementata, le pratiche di sicurezza hanno un forte effetto positivo sulle prestazioni di distribuzione del software. Ciò significa essenzialmente che la CI è necessaria perché le pratiche di sicurezza abbiano un impatto positivo sulle prestazioni di distribuzione del software. Inoltre, le pratiche di sicurezza hanno generalmente un effetto positivo sulle prestazioni dell'organizzazione e, quando la CI è saldamente implementata, questo effetto viene amplificato. Il grafico sottostante tenta di visualizzare questo effetto.





Insieme a una riduzione dei rischi per la sicurezza percepiti, gli intervistati hanno riferito anche un **minor burnout** tra i membri del team e una maggiore disponibilità a consigliare la propria organizzazione come **un ottimo posto in cui lavorare**. Entrambi questi risultati denotano la natura collaborativa della sicurezza per gli ingegneri del software: è un'attività che si aggiunge alle tante altre di cui occuparsi. Gli strumenti e i processi che li aiutano a incorporare pratiche sicure nel loro flusso di lavoro di sviluppo esistente, al contrario del lavoro non pianificato o delle "esercitazioni antincendio" quando viene scoperta una minaccia, forniscono un meccanismo per ridurre i rischi per la sicurezza e aumentare la soddisfazione degli sviluppatori.

Nel loro insieme, i nostri riscontri suggeriscono che i team **sani e con prestazioni elevate tendono anche ad avere buone pratiche di sicurezza** ampiamente implementate (sebbene, come notato in precedenza, continuino ad esserci margini di miglioramento). Seguire approcci come i framework SLSA o SSDF potrebbe non bastare a migliorare tutte le metriche relative alla cultura e alle prestazioni che misuriamo, ma è chiaro che la sicurezza non deve andare a scapito di altre priorità di sviluppo.

Gli strumenti e i processi che li aiutano a incorporare pratiche sicure nel loro flusso di lavoro di sviluppo esistente, al contrario del lavoro non pianificato o delle "esercitazioni antincendio" quando viene scoperta una minaccia, forniscono un meccanismo per ridurre i rischi per la sicurezza e aumentare la soddisfazione degli sviluppatori.

# 05

## Sorprese



Derek DeBellis

Sebbene il report di ogni anno si concentri sulle risposte al sondaggio dell'anno corrispondente, facciamo del nostro meglio per comprendere questi risultati nel contesto dell'intero catalogo dei report State of DevOps e delle ricerche limitrofe (ad esempio, la ricerca sul burnout e la cultura). Testare l'affidabilità di questi effetti attraverso iniziative di replica è un principio fondamentale del programma di ricerca. Questo ci dà l'opportunità di modificare le nostre convinzioni per adattarle ai dati e comprendere le tendenze in evoluzione o emergenti.

Quest'anno abbiamo avuto qualche sorpresa. C'è una miriade di potenziali ragioni per questo. Per prima cosa, quest'anno il campione è passato a includere più persone all'inizio della loro carriera rispetto ai report precedenti. Una interpretazione è che stiamo intervistando di più le persone che sono direttamente responsabili dell'implementazione delle pratiche e delle funzionalità tecniche, rispetto alle persone che

potrebbero essere responsabili della supervisione o della direzione nell'attuazione di queste pratiche. Un'altra possibilità è che qualcosa sia cambiato nel settore o nel mondo; ciò che ha funzionato ieri non è garantito che funzioni domani. Le forze macroeconomiche e un altro anno in gran parte all'ombra della pandemia di COVID-19, ad esempio, potrebbero aver cambiato la fisica delle DevOps. Infine, sottili modifiche a ciò che è incluso nel nostro modello potrebbero aver modificato le relazioni tra le variabili.<sup>1</sup>



<sup>1</sup> "Book of Why" di Judea Pearl e "Statistical Rethinking" di Robert McElreath ci presentano incredibili esempi di come ciò che includi o non includi nei tuoi modelli statistici può influire sull'output del modello.

Una scoperta inaspettata o non ipotizzata mette i ricercatori in una situazione difficile quando è il momento di scrivere il report. Dato il rischio che la scoperta sia falsa o, per lo meno, debba ancora confermare (o addirittura contraddirsi) l'evidenza empirica di diversi studi, la mossa più responsabile è fare ricerche di verifica per tentare di replicare i risultati e capirne la causa.<sup>2</sup> Se si concentra sulle sorprese, un ricercatore corre anche il rischio di sottovalutare quanti effetti sono emersi in modo affidabile in anni di ricerca. Esploriamo statisticamente oltre cento percorsi per ogni report State of DevOps. In questo modo invitiamo il rischio di risultati fuorvianti ottenuti semplicemente per caso. Cerchiamo di contrastare questa evenienza con iniziative di replica annuali.

D'altra parte, la mancata comunicazione del ritrovamento rischia di creare un effetto schedario<sup>3</sup>, che di fatto fa sì che ciò che è atteso o appetibile diventi noto e ciò che è inaspettato o difficile da digerire venga nascosto. Cerchiamo di trovare un equilibrio: non vogliamo sensazionalizzare le scoperte nascenti, ma riteniamo anche che sia fondamentale condividerle. Ecco le cose che ci hanno sorpreso di più e cosa pensiamo significhino:

**01** Abbiamo costantemente osservato che le pratiche di sviluppo basate su trunk hanno un impatto positivo sulle prestazioni di distribuzione del software. In effetti, questo è stato osservato in ogni anno dello studio dal 2014. Le funzionalità di sviluppo basate su trunk si sono comportate in modo insolito quest'anno. Ad esempio, le funzionalità trunk hanno avuto un impatto negativo sulle prestazioni di distribuzione del software, contrariamente a quanto accadeva nei report di ricerca precedenti. Dato quanto è aberrante questo risultato, non vediamo l'ora di vedere se verrà replicato nelle prossime ricerche e sentire se la community ha qualche spiegazione.

**02** Abbiamo scoperto che le prestazioni di distribuzione del software sono vantaggiose per le prestazioni dell'organizzazione solo quando anche le prestazioni operative sono elevate, e molti intervistati non avevano prestazioni operative elevate. Ciò contraddice le precedenti iterazioni della nostra ricerca in cui la connessione tra le prestazioni di distribuzione del software e le prestazioni dell'organizzazione era molto più chiara.

<sup>2</sup> Kerr, N. L. (1998). HARKing: Hypothesizing after the results are known. *Personality and social psychology review*, 2(3), 196-217.

<sup>3</sup> Rosenthal, R. (1979). The file drawer problem and tolerance for null results. *Psychological bulletin*, 86(3), 638.

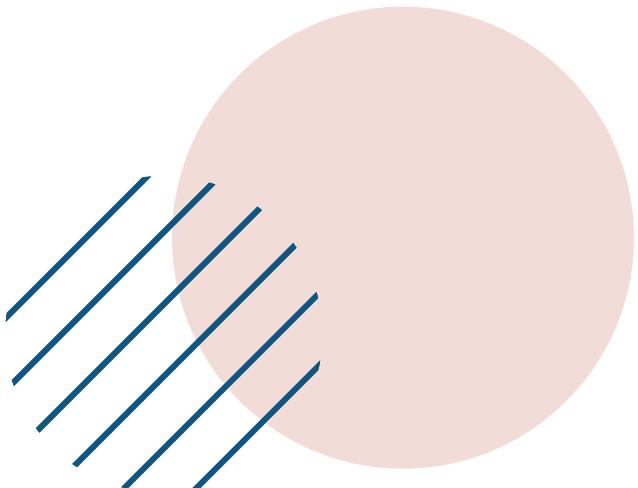
- 
- 03** Le pratiche relative alla documentazione hanno avuto un impatto negativo sulle prestazioni di distribuzione del software. Questo è in contrasto con i report precedenti. Un'ipotesi è che la documentazione stia diventando una pratica sempre più automatizzata, soprattutto tra i team con prestazioni elevate. Fino a quando non raccoglieremo ulteriori dati non avremo riscontri sufficienti per supportare o confutare questa convinzione.
- 04** Alcune funzionalità tecnologiche (ad es. sviluppo basato su trunk, architettura a basso accoppiamento, CI, CD) sembrano far prevedere il burnout. Come accennato in precedenza, molti intervistati in questo campione erano notevolmente più all'inizio della loro carriera rispetto ai partecipanti dei campioni degli anni precedenti. Quindi, potremmo aver parlato con le persone responsabili dell'implementazione della funzionalità piuttosto che con quelle responsabili della creazione o della supervisione dell'iniziativa. Il processo di implementazione può essere notevolmente più impegnativo della sua supervisione. Vogliamo fare ulteriori ricerche per comprendere meglio questo risultato.

- 05** Le pratiche di ingegneria dell'affidabilità hanno avuto un impatto negativo sulle prestazioni di distribuzione del software. Una spiegazione è che i due elementi non sono necessariamente intrecciati in modo causale. Quest'anno abbiamo notato in una nuova analisi di clustering (vedi "Come ti collochi?") che un sottoinsieme di cluster sembrava concentrarsi sull'affidabilità ignorando le prestazioni di distribuzione del software. Riteniamo che le due siano disaccoppiate, nel senso che puoi dedicarti a una senza pensare all'altra, ma alla fine, per fare in modo che le prestazioni di distribuzione del software contino in termini di prestazioni dell'organizzazione, l'affidabilità deve essere presente.
- 06** Abbiamo aggiunto pratiche relative al framework SLSA per capire se i team stiano adottando questi approcci per garantire la sicurezza della catena di fornitura del software. Mentre ci aspettavamo che ci fosse un'associazione tra l'implementazione delle pratiche di sicurezza e le prestazioni (ad es. utilizzo di funzionalità tecniche, migliori prestazioni di distribuzione del software e migliori prestazioni dell'organizzazione), ci ha sorpreso riscontrare che le pratiche di sicurezza erano in realtà il meccanismo attraverso il quale le funzionalità tecniche influivano sulle prestazioni di distribuzione del software e sulle prestazioni dell'organizzazione.

L'inclusione di pratiche relative al framework SLSA sembra spiegare gran parte dell'effetto dell'integrazione continua, del controllo della versione e della distribuzione continua sia sulle prestazioni di distribuzione del software che sulle prestazioni dell'organizzazione. In altre parole, c'è una catena causale che viene rilevata nei dati per cui molte funzionalità tecniche hanno un impatto positivo sulle pratiche relative al framework SLSA e di conseguenza anche sulle prestazioni di distribuzione del software e sulle prestazioni dell'organizzazione. Abbiamo utilizzato analisi di mediazione per rilevare questo risultato.<sup>45</sup> Questo ci sta spingendo a esplorare se la nostra misura delle pratiche relative a SLSA tiene conto di altre caratteristiche del team (ad esempio, prestazioni generali) e in che modo le pratiche di sicurezza portano a migliori prestazioni di distribuzione del software e dell'organizzazione.

Non vediamo l'ora di studiare nuovamente questi effetti l'anno prossimo, per vedere se possiamo riprodurre e spiegare questi nuovi modelli, o se dovremmo tendere a liquidarli come valori anomali (che dovremmo anche cercare di spiegare). Come sempre, accogliamo con favore il feedback della community.

Unisciti alla [DORA Community](http://dora.community) (<http://dora.community>) per continuare la discussione su queste sorprese e altri risultati nel report di quest'anno!



<sup>4</sup> Jung, Sun Jae. "Introduction to Mediation Analysis and Examples of Its Application to Real-world Data." Journal of preventive medicine and public health = Yebang Uihakhoe chi vol. 54,3 (2021): 166-172. doi:10.3961/jpmph.21.069

<sup>5</sup> Carrión, Gabriel Cepeda, Christian Nitzl e José L. Roldán. "Mediation analyses in partial least squares structural equation modeling: Guidelines and empirical examples." Partial least squares path modeling. Springer, Cham, 2017. 173-195.

# 06

## Informazioni demografiche e aziendali



Derek DeBellis

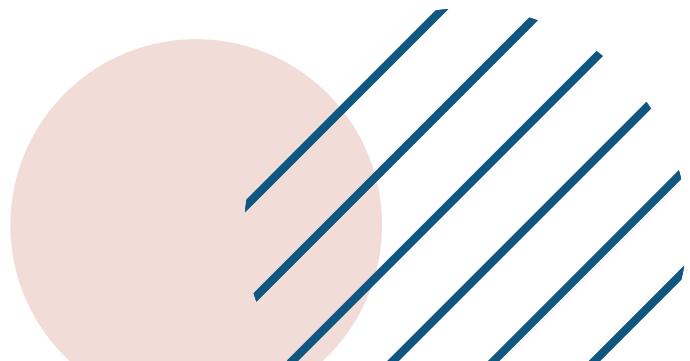
Grazie per il contributo apportato alla nostra ricerca e al nostro settore.

### Chi ha partecipato al sondaggio?

Con otto anni di ricerca e oltre 33.000 risposte ai sondaggi da parte di professionisti del settore, il report State of DevOps mostra quali sono le pratiche di sviluppo software e DevOps che contribuiscono maggiormente al successo dei team e delle organizzazioni. Quest'anno, 1350 professionisti di una varietà di settori in tutto il mondo hanno condiviso le loro esperienze per contribuire ad aumentare la nostra comprensione dei fattori che determinano prestazioni più elevate. Grazie per il contributo apportato alla nostra ricerca e al nostro settore. Per riassumere, la rappresentazione di gruppi demografici e aziendali è rimasta notevolmente coerente.

Analogamente agli anni precedenti, abbiamo raccolto dati demografici per ogni partecipante al sondaggio. Le categorie includono genere, disabilità e gruppi sottorappresentati.

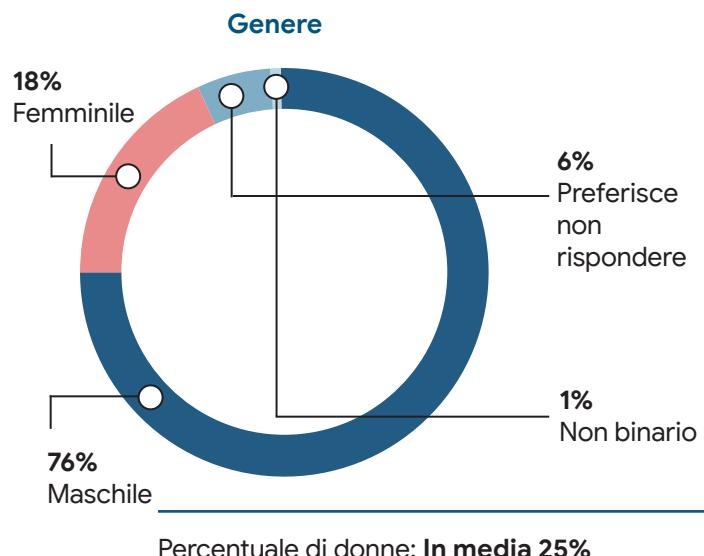
Quest'anno abbiamo visto una rappresentazione coerente con i report precedenti nelle categorie delle informazioni aziendali, tra cui dimensioni dell'azienda, settore e area geografica. Di nuovo, oltre il 60% degli intervistati è composto da tecnici o manager e un terzo lavora nel settore tecnologico. Inoltre, vediamo rappresentati i settori dei servizi finanziari, retail e delle aziende industriali/di produzione.



# Dati demografici

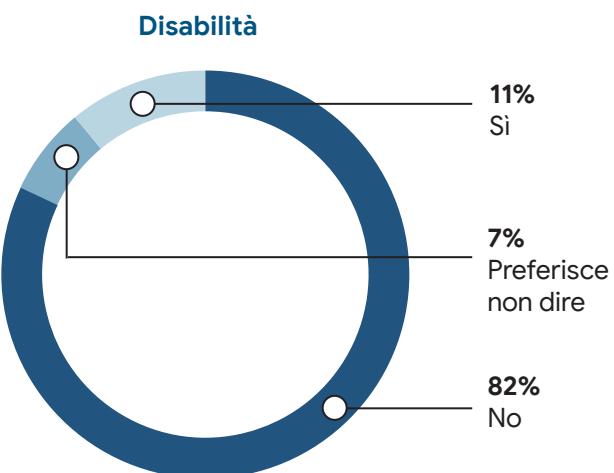
## Genere

Rispetto al 2021, il campione di quest'anno ha avuto una percentuale più alta di intervistate donne (il 18% contro il 12%). La percentuale di intervistati di sesso maschile (76%) è stata inferiore rispetto al 2021 (83%). Gli intervistati hanno affermato che le donne costituiscono il 25% dei loro team, dato identico al 2021 (25%).



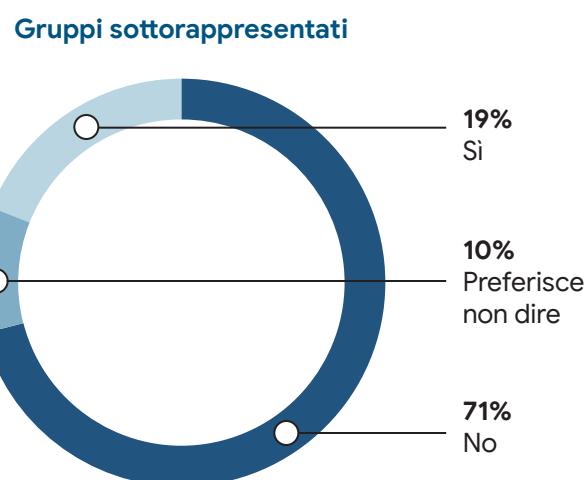
## Disabilità

La disabilità è identificata lungo sei dimensioni che seguono le linee guida del [Washington Group Short Set](#). Questo è il quarto anno in cui abbiamo posto domande sulla disabilità. La percentuale di persone con disabilità è rimasta stabile all'11% rispetto al nostro report del 2021.



## Gruppi sotorappresentati

L'identificazione come appartenente a un gruppo sotorappresentato può far riferimento al gruppo etnico, al genere o a un'altra caratteristica. Questo è il quinto anno in cui abbiamo posto domande sui gruppi sotorappresentati. La percentuale di persone che si identificano come sotorappresentate è aumentata leggermente, dal 17% del 2021 al 19% del 2022.

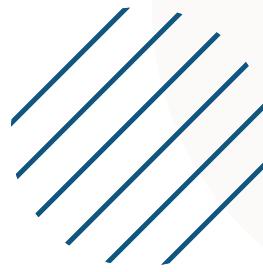


## Anni di esperienza

In particolare, più intervistati quest'anno avevano cinque anni o meno di esperienza (35%) rispetto al 2021 (14%). Forse non sorprende quindi che la percentuale di intervistati con più di 16 anni di esperienza (13%) fosse una frazione rispetto al 2021 (41%).

Questo spostamento potrebbe spiegare alcuni modelli emersi nei dati, e crediamo sia importante tenerlo a mente quando si interpretano i risultati, soprattutto quando si confrontano con lo scorso anno





# Informazioni aziendali

## Ruolo

L'85% degli intervistati è costituito da persone che lavorano in team di sviluppo o progettazione (26%), in team DevOps o SRE (23%), in team delle operazioni IT o dell'infrastruttura (19%) o sono manager (17%). La percentuale di intervistati che lavorano in team delle operazioni IT o dell'infrastruttura (19%) è più che raddoppiata rispetto allo scorso anno (9%). Quelle degli intervistati di livello dirigenziale (dal 9% nel 2021 al 4%) e appartenenti a servizi professionali (dal 4% nel 2021 all'1%) sono due delle diminuzioni più pronunciate rispetto allo scorso anno.



## 2022

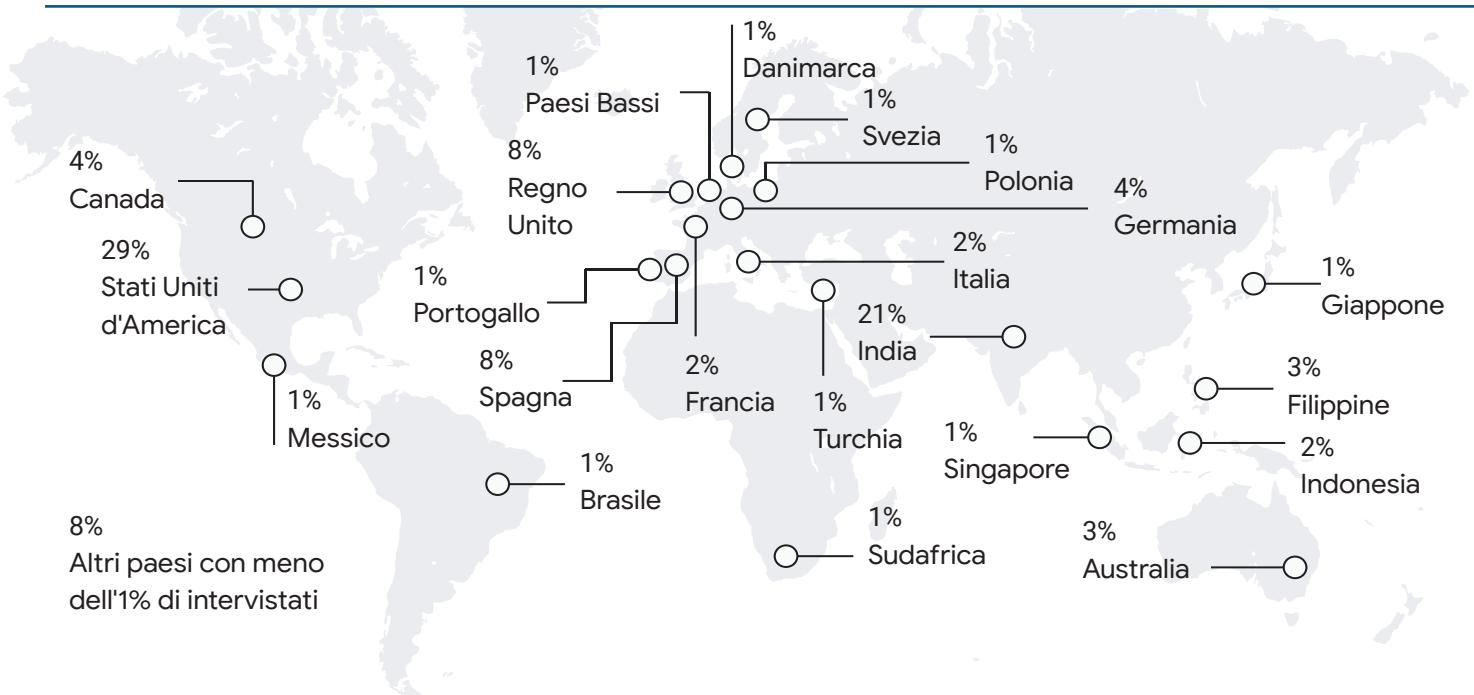


## Settore

Come nei precedenti report State of DevOps, vediamo che la maggior parte degli intervistati lavora nel settore tecnologico, seguito da servizi finanziari, altro e vendita al dettaglio.

## Regione

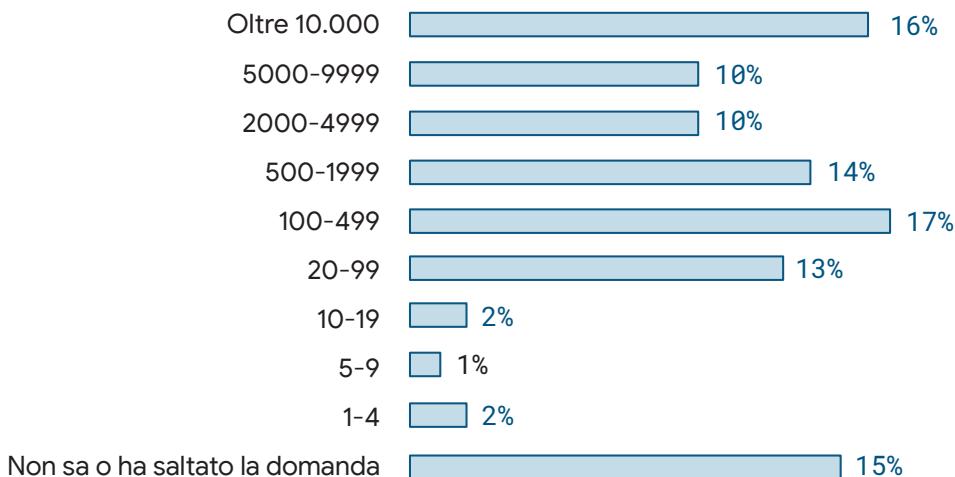
Quest'anno abbiamo chiesto agli intervistati di selezionare il paese di provenienza anziché la regione. La regione, spesso rappresentata da un continente, sembrava un po' troppo grossolana per comprendere la composizione dei nostri intervistati. Abbiamo ricevuto risposte dai partecipanti in oltre 70 paesi. L'89% degli intervistati proveniva da 22 paesi.

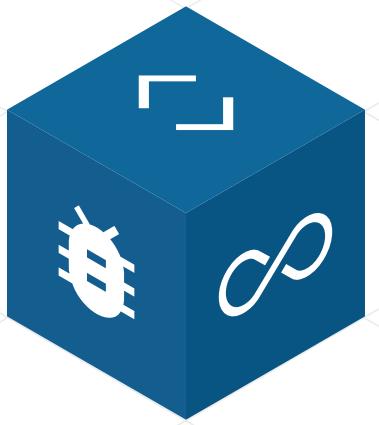


## Numero di dipendenti

Coerentemente con i precedenti sondaggi State of DevOps, gli intervistati provengono da organizzazioni di varie dimensioni. Il 22% degli intervistati lavora in aziende con oltre 10.000 dipendenti e il 7% in aziende con 5000-9999 dipendenti. Un altro 15% degli intervistati lavora in

organizzazioni con 2000-4999 dipendenti. Abbiamo anche visto una buona rappresentanza di intervistati (13%) provenienti da organizzazioni con 500-1999 dipendenti, un 15% con 100-499 dipendenti e infine un 15% con 20-99 dipendenti. Quest'anno, abbiamo anche consentito agli intervistati di selezionare "Non so" in merito alle dimensioni della loro organizzazione. Il 15% degli intervistati ha dichiarato di non sapere o ha saltato la domanda.

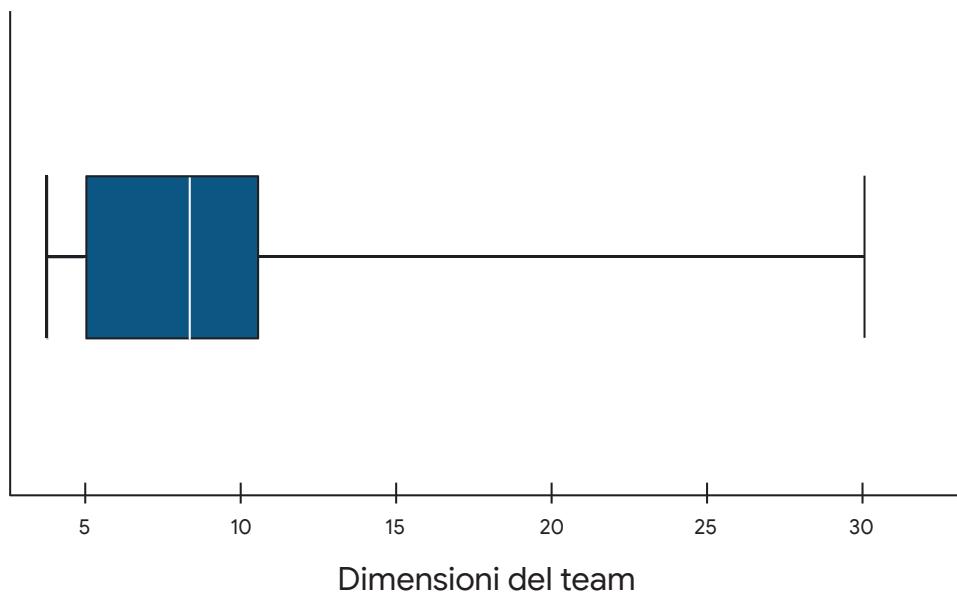




## Dimensioni del team

Quest'anno abbiamo chiesto ai partecipanti di indicare il numero approssimativo di persone nel loro team.

Il 25% degli intervistati faceva parte di team con 5 persone o meno. Il 50% degli intervistati faceva parte di team con 8 persone o meno. Il 75% degli intervistati faceva parte di team con 12 persone o meno.

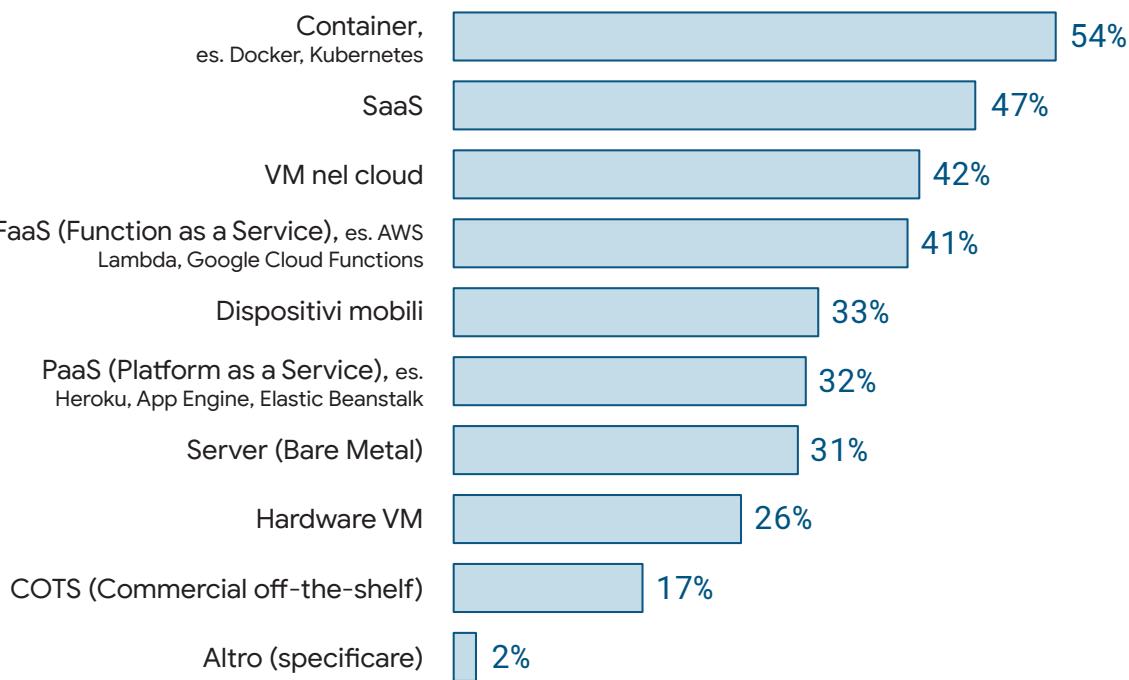


## Destinazione di deployment

Il 2021 è stato il primo anno in cui abbiamo deciso di esaminare dove gli intervistati hanno eseguito il deployment del servizio o dell'applicazione principale su cui lavorano. Con nostra sorpresa, la prima destinazione del deployment

sono risultati i container. Quest'anno non è stato diverso, anche se con una percentuale inferiore (54%) rispetto allo scorso anno (64%).

Abbiamo anche aggiunto più opzioni nella speranza di offrire agli intervistati più modi per riflettere su dove eseguire il deployment.



# 07

## Considerazioni finali



Derek DeBellis

Ogni anno in cui pubblichiamo questo report, ci sforziamo di fornire un resoconto rigoroso di come le pratiche e le funzionalità portano a risultati business-critical come le prestazioni dell'organizzazione. Valutiamo la replicabilità di molti effetti riscontrati nei report precedenti ed estendiamo l'ambito del nostro lavoro per tenere conto delle priorità emergenti nell'area DevOps. Quest'anno abbiamo modellato il nostro sondaggio e le nostre analisi per approfondire le pratiche di sicurezza e abbiamo modificato il nostro approccio di modellazione statistica per esplorare la condizionalità o le dipendenze di determinati effetti. Abbiamo anche esplorato nuovi modi per descrivere il panorama della distribuzione del software e delle prestazioni operative.

Per molti versi, la narrazione che si è materializzata quest'anno è un'eco degli anni precedenti: le funzionalità tecniche si basano l'una sull'altra per creare prestazioni migliori; ci sono molti vantaggi inerenti all'uso del cloud; la cultura e la flessibilità del posto di lavoro portano a migliori prestazioni dell'organizzazione; il burnout dei dipendenti impedisce alle organizzazioni di raggiungere i propri obiettivi. L'analisi delle interazioni che abbiamo esplicitamente aggiunto al modello ci ha aiutato a capire le condizioni in cui possono

verificarsi determinati effetti. Ad esempio, le prestazioni di distribuzione del software sembrano avere un impatto positivo sulle prestazioni dell'organizzazione solo quando le prestazioni operative (affidabilità) sono elevate, il che porta alla conclusione che sono necessarie entrambe per prosperare come organizzazione. Non sono mancate anche alcune sorprese, che abbiamo evidenziato in una sezione dedicata.

Ringraziamo tutti coloro che hanno contribuito al sondaggio di quest'anno. Ci auguriamo che la nostra ricerca aiuti te e la tua organizzazione a creare team migliori e software migliori, mantenendo al contempo l'equilibrio tra lavoro e vita privata.

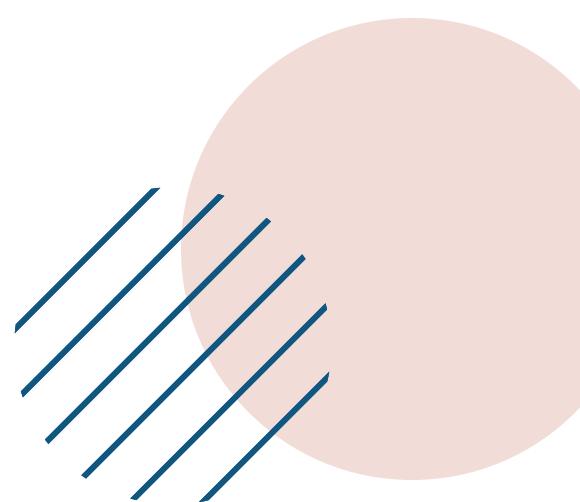


# 08

## Ringraziamenti

Una grande famiglia di collaboratori appassionati ha reso possibile la realizzazione del report di quest'anno. La progettazione, l'analisi, la scrittura, la modifica delle domande del sondaggio e la progettazione del report sono solo alcuni dei modi in cui i nostri colleghi ci hanno aiutato a portare a compimento questo grande impegno. Gli autori vogliono cogliere l'occasione per ringraziare tutte queste persone per il loro contributo e la loro guida al report di quest'anno. Tutti i riconoscimenti sono in ordine alfabetico.

Scott Aucoin	Eric Maxwell
Alex Barrett	John Speed Meyers
James Brookbank	Steve McGhee
Kim Castillo	Jacinda Mein
Lolly Chessie	Alison Milligan
Jenna Dailey	Pablo Pérez Villanueva
Derek DeBellis	Claire Peters
Rob Edwards	Connor Poske
Dave Farley	Dave Stanke
Christopher Grant	Dustin Smith
Mahshad Haeri	Seth Vargo
Nathen Harvey	Daniella Villalba
Damith Karunaratne	Brenna Washington
Todd Kulesza	Kaiyuan "Frank" Xu
Amanda Lewis	Nicola Yap
Ian Lewis	



# 09

## Autori



### Claire Peters

Claire Peters è User Experience Researcher presso Google. La sua ricerca comprende vari aspetti ed espressioni di DORA in contesti applicati. Studia il Cloud Application Modernization Program (CAMP) di Google, il coinvolgimento dei clienti incentivato da DORA e gli strumenti relativi a Four-Keys, con l'obiettivo di aiutare i team e le persone ad applicare in modo più efficace i principi DORA nel loro lavoro quotidiano. Claire è anche un membro del team di ricerca principale di DORA e realizza il sondaggio annuale DORA e il report State of DevOps. Ha conseguito un MS in Analisi Culturale Applicata presso l'Università di Copenhagen.



### Dave Farley

Dave Farley, è l'amministratore delegato e fondatore di Continuous Delivery Ltd e creatore del canale YouTube Continuous Delivery. Dave è coautore del best-seller Continuous Delivery. È anche tra gli autori più venduti con Modern Software Engineering: Doing What Works to Build Better Software Faster. È coautore del Reactive Manifesto e vincitore del Duke Award per il progetto open source LMAX Disruptor. Dave è un pioniere della distribuzione continua, leader di pensiero e professionista esperto in CD, DevOps, TDD e progettazione di software, e ha una lunga esperienza nella creazione di team con prestazioni elevate, nel condurre le organizzazioni al successo e nella creazione di software eccezionale. Trova altre informazioni su Dave su: [Twitter](#), [YouTube](#), [il suo blog](#) e [il suo sito web](#).



## Daniella Villalba

Daniella Villalba è una ricercatrice che si occupa di esperienza utente e si dedica al progetto DORA. Si concentra sulla comprensione dei fattori che rendono gli sviluppatori contenti e produttivi. Prima di entrare in Google, Daniella ha studiato i vantaggi dell'addestramento alla meditazione e i fattori psico-sociali che influiscono sulle esperienze degli studenti universitari, sulla memoria dei testimoni oculari e sulle false confessioni. Ha conseguito il PhD in psicologia sperimentale presso la Florida International University.



## Dave Stanke

Dave Stanke lavora come Developer Relations Engineer in Google, dove consiglia ai clienti le pratiche per l'adozione di DevOps e SRE. Nell'arco della sua carriera ha ricoperto ogni genere di posizione, tra cui CTO di una startup, product manager, assistenza clienti, sviluppatore software, amministratore di sistema e graphic designer. Ha conseguito un MS in gestione della tecnologia presso la Columbia University.



## Derek DeBellis

Derek DeBellis è Quantitative User Experience Researcher in Google, dove concentra il suo lavoro sulla ricerca tramite sondaggi, sull'analisi dei log e sulla scoperta di modi per misurare i concetti fondamentali per lo sviluppo del prodotto. Derek ha recentemente pubblicato un articolo sull'interazione uomo-intelligenza artificiale, sull'impatto dell'insorgenza di Covid-19 sulla cessazione del fumo, sulla progettazione per gli errori di NLP e sul ruolo dell'esperienza utente nelle discussioni sulla privacy.



## Eric Maxwell

Eric Maxwell guida la pratica di trasformazione digitale DevOps di Google, dove consiglia le migliori aziende del mondo su come essere ancora migliori, un po' alla volta, in modo continuativo. Eric ha trascorso la prima metà della sua carriera come ingegnere, in trincea, automatizzando di tutto e creando empatia per gli altri colleghi. Eric ha co-creato il Cloud Application Modernization Program (CAMP) di Google, è membro del DORA Core Team e autore della DevOps Enterprise Guidebook. Prima di Google, Eric ha trascorso del tempo a creare fantastiche cose con altre persone argute di Chef Software.



## John Speed Meyers

John Speed Meyers è Security Data Scientist presso Chainguard, una startup per la sicurezza della catena di fornitura di software. I progetti di ricerca di John hanno incluso argomenti come la sicurezza della catena di fornitura del software, la sicurezza del software open source e la risposta del mondo alla crescente potenza militare cinese. In precedenza, John ha lavorato presso In-Q-Tel, la RAND Corporation e il Center for Strategic and Budgetary Assessments. John ha conseguito un PhD in analisi delle politiche presso la Pardee RAND Graduate School, un MPA presso la School of Public and International Affairs di Princeton e un BA in relazioni internazionali presso la Tufts University.



## Kaiyuan "Frank" Xu

Kaiyuan "Frank" Xu è Quantitative User Experience Researcher presso Google. Analizza i dati di log e sondaggi per comprendere i modelli di utilizzo e il feedback degli utenti per i prodotti Google Cloud, migliorando l'eccellenza del prodotto per gli sviluppatori. Prima di Google, Kaiyuan ha condotto anni di ricerca qualitativa e quantitativa sugli utenti presso Microsoft per i prodotti Azure e Power Platform. Ha conseguito la laurea specialistica in Human Centered Design and Engineering presso l'Università di Washington.



## Nathen Harvey

Nathen Harvey lavora come Developer Relations Engineer in Google e ha costruito la sua carriera sull'aiuto ai team nel realizzare il loro potenziale e al contempo allineare la tecnologia ai risultati aziendali. Nathen ha avuto il privilegio di lavorare con alcuni dei migliori team e delle migliori community open source, aiutandoli ad applicare principi e pratiche di DevOps e SRE. Nathen è coredattore e ha contribuito alla pubblicazione del libro *97 Things Every Cloud Engineer Should Know*, O'Reilly 2020.



## Todd Kulesza

Todd Kulesza è User Experience Researcher presso Google, dove studia come lavorano gli ingegneri informatici oggi e come potrebbero essere in grado di lavorare meglio domani. Ha conseguito un PhD in Informatica presso la Oregon State University.

# 10

## Metodologia

### Progettazione della ricerca

Questo studio utilizza un tipo di progettazione trasversale basato sulla teoria, noto come **predittivo-inferenziale**, uno dei tipi più comuni di progettazione condotti oggi nella ricerca aziendale e tecnologica. La progettazione predittivo-inferenziale viene utilizzata quando la progettazione puramente sperimentale non è pratica o è impossibile.

### Popolazione target e campionamento

La popolazione target per questo sondaggio era costituita da professionisti e leader che lavorano direttamente, o in stretto contatto, con la tecnologia e le trasformazioni, in particolare quanti hanno dimestichezza con le DevOps. Abbiamo promosso il sondaggio tramite mailing list, promozioni online, un gruppo online, social media, oltre che chiedendo di condividere il sondaggio con le proprie reti (il cosiddetto campionamento a valanga).

### Creazione dei costrutti latenti

Abbiamo formulato ipotesi e costrutti utilizzando, dove possibile, costrutti convalidati in precedenza. Abbiamo sviluppato nuovi costrutti in base alla teoria, alle definizioni e agli input degli esperti. Abbiamo intrapreso ulteriori passaggi per chiarire l'intento, al fine di assicurare che i dati raccolti dal sondaggio avessero un'alta probabilità di essere affidabili e validi.<sup>1</sup>

### Calcolo delle differenze tra prestazioni basse ed elevate

Nella sezione "Come ti collochi?" mettiamo a confronto gli intervistati con prestazioni basse e quelli con prestazioni elevate sulle quattro metriche relative alle prestazioni di distribuzione. Il metodo è semplice. Prendiamo come esempio la frequenza di deployment. Il cluster alto esegue il deployment on demand (ovvero più volte al giorno). Se eseguono una media di quattro deployment al giorno, si ottengono 1460 deployment all'anno ( $4 * 365$ ). Gli intervistati con prestazioni basse, al contrario, effettuano il deployment da una volta al mese a una volta ogni sei mesi, con una frequenza media di deployment di una volta ogni 3,5 mesi e una frequenza media di circa 3,4 deployment all'anno ( $12/3,5$ ). Prendiamo il rapporto (1460 deployment alta / 3,4 deployment bassa) e otteniamo 417 volte. Questo approccio è generalizzato per le altre metriche sulle prestazioni dello sviluppo.

<sup>1</sup> Churchill Jr, G. A. "A paradigm for developing better measures of marketing constructs", Journal of Marketing Research 16:1, (1979), 64–73.

# Metodi di analisi statistica

## Cluster analysis

Per entrambe le soluzioni di clustering descritte nella sezione "Come ti collochi?", abbiamo utilizzato il clustering gerarchico con una versione del metodo agglomerativo di Ward<sup>2</sup> per valutare in che modo le diverse soluzioni di cluster si adattano ai dati.

Per i primi risultati di clustering che abbiamo presentato, abbiamo cercato cluster di risposte per frequenza di implementazione, tempo di risposta, tempo per ripristinare un servizio e tasso di errore delle modifiche. Quest'anno abbiamo trovato tre cluster dopo aver valutato 14 diverse soluzioni di clustering gerarchico utilizzando 30 diversi indici per determinare il numero di cluster.<sup>3</sup>

La seconda cluster analysis che abbiamo presentato era metodologicamente identica alla prima, ma è stata distribuita su diverse dimensioni dei dati. Volevamo cercare modelli di risposta comuni (ad es. cluster) per la

velocità effettiva (un insieme di frequenza di deployment e tempo di risposta), le prestazioni operative (affidabilità) e la stabilità (un insieme di tempo per ripristinare il servizio e tasso di errore delle modifiche). Abbiamo anche esplorato diversi algoritmi di clustering per vedere quanto i nostri risultati fossero sensibili al nostro approccio. Sebbene non esista un modo consolidato per quantificare questa sensibilità (di cui siamo a conoscenza), i cluster emersi tendevano ad avere caratteristiche simili.

## Modello di misurazione

Prima di condurre la nostra analisi, abbiamo identificato i costrutti utilizzando l'analisi fattoriale esplorativa con l'analisi delle componenti principali utilizzando la rotazione Varimax.<sup>4</sup> Abbiamo confermato i test statistici per la validità e l'affidabilità convergenti e divergenti utilizzando varianza media estratta (AVE), correlazione, alpha di Cronbach<sup>5</sup>, rhoA<sup>6</sup>, rapporto HTMT (heterotrait-monotrait)<sup>5,7</sup> e affidabilità composita.

<sup>2</sup> Murtagh, Fionn e Pierre Legendre. "Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?." *Journal of classification* 31.3 (2014): 274-295.

<sup>3</sup> Charrad M., Ghazzali N., Boiteau V., Niknafs A. (2014). "NbClust: An R Package for Determining the Relevant Number of Clusters in a Data Set.", "Journal of Statistical Software, 61(6), 1-36.", "URL <http://www.jstatsoft.org/v61/i06/>"

<sup>4</sup> Straub, D., Boudreau, M. C., & Gefen, D. (2004). Validation guidelines for IS positivist research. *Communications of the Association for Information systems*, 13(1), 24.

<sup>5</sup> Nunnally, J.C. *Psychometric Theory*. New York: McGraw-Hill, 1978

<sup>6</sup> Hair Jr, Joseph F., et al. "Partial least squares structural equation modeling (PLS-SEM) using R: A workbook." (2021): 197.

<sup>7</sup> Brown, Timothy A. e Michael T. Moore. "Confirmatory factor analysis." *Handbook of structural equation modeling* 361 (2012): 379.

## **Modellazione di equazioni strutturali**

Abbiamo testato i modelli di equazioni strutturali (SEM) utilizzando la tecnica dei minimi quadrati parziali (PLS)<sup>8</sup>, che consiste in una modellazione di equazioni strutturali basata sulla correlazione.

## **Analisi del secondo modello di cluster**

Per capire cosa può prevedere l'appartenenza al cluster, abbiamo utilizzato la regressione logistica multinomiale.<sup>9</sup> Abbiamo utilizzato questo approccio perché volevamo prevedere l'appartenenza al cluster, che, in questo caso, sono dati categorici non ordinati con più di due livelli. Per comprendere i risultati previsti dall'appartenenza al cluster, abbiamo utilizzato una regressione lineare per ciascun risultato (burnout, lavoro non pianificato e prestazioni dell'organizzazione).

<sup>8</sup> Hair Jr, J. F., Hult, G. T. M., Ringle, C. M., & Sarstedt, M. (2021). "A primer on partial least squares structural equation modeling (PLS-SEM)." Sage publications.

<sup>9</sup> Ripley, Brian, William Venables e Maintainer Brian Ripley. "Package 'nnet'." R package version 7.3-12 (2016): 700.

# 11

## Per approfondire

Unisciti alla DORA Community per discutere, apprendere e collaborare per migliorare le prestazioni operative e di distribuzione del software  
<http://dora.community>

Scopri di più sulle metriche delle "quattro chiavi"  
<https://goo.gle/four-keys>

Scopri di più sulle funzionalità DevOps  
<https://goo.gle/devops-capabilities>

Scopri di più su come implementare le pratiche DORA nella tua organizzazione, dalla nostra Enterprise Guidebook  
<https://goo.gle/enterprise-guidebook>

Trova risorse sull'SRE (Site Reliability Engineering) all'indirizzo  
<https://sre.google>  
<https://goo.gle/enterprise-roadmap-sre>

Esegui il controllo rapido delle DevOps  
<https://goo.gle/devops-quickcheck>

Esplora il programma di ricerca sulle DevOps  
<https://goo.gle/devops-research>

Impara da altre aziende che hanno implementato le pratiche DORA leggendo l'ebook vincitore del Google Cloud DevOps Award  
<https://goo.gle/devops-awards>

Scopri il Cloud Application Modernization Program (CAMP) di Google  
<https://goo.gle/3daLa9s>

Sfrutta i dati e le metriche DORA per trasformare i processi tecnologici  
<https://goo.gle/3Doh8Km>

Leggi il white paper: "The ROI of DevOps Transformation: How to quantify the impact of your modernization initiatives" di Forsgren, N., Humble, J., & Kim, G. (2018)  
<https://goo.gle/3qECIIh>

Leggi il libro: Accelerate: *The science behind devops: Building and scaling high performing technology organizations.* IT Revolution  
<https://itrevolution.com/book/accelerate>

Scopri di più sul framework SLSA (Supply chain Levels for Secure Artifacts)  
<https://slsa.dev>

Scopri di più su Secure Software Development Framework (NIST SSDF)  
<https://goo.gle/3qBXLWk>

Scopri di più sulla cultura DevOps: la cultura dell'organizzazione di Westrum  
<https://goo.gle/3xq7KBV>

Scopri di più sulla Open Source Security Foundation  
<https://openSSF.org/>

Scopri di più su in-toto  
<https://in-toto.io/>

Scopri di più sulla Software Bill of Materials di NTIA.gov  
<https://www.ntia.gov/SBOM>

Cybersecurity: Federal Response to SolarWinds and Microsoft Exchange Incidents  
<https://www.gao.gov/products/gao-22-104746>

Scopri di più sulla scansione di sicurezza a livello di applicazione come parte del sistema CI/CD  
<https://go.dev/blog/survey2022-q2-results#security>

Ultimo ma non in ordine di importanza, consulta i precedenti report State of DevOps. Sono tutti disponibili nell'elenco su <https://goo.gle/dora-sodrs>:  
[Report 2014 Accelerate State of DevOps](#)  
[Report 2015 Accelerate State of DevOps](#)  
[Report 2016 Accelerate State of DevOps](#)  
[Report 2017 Accelerate State of DevOps](#)  
[Report 2018 Accelerate State of DevOps](#)  
[Report 2019 Accelerate State of DevOps](#)  
[Report 2021 Accelerate State of DevOps](#)

