Necessità a livello di business di ridurre il time-to-market (definizione time to market): oggi si chiedono tempi strettissimi di sviluppo per abilitare rapidamente la disponibilità di nuovi prodotti e servizi. Rilasci sempre più piccoli e frequenti. DevOps 🡪 maggiore competitività sul mercato

La complessità delle strutture e infrastrutture It ha alimentato negli ultimi anni conflitti inter-organizzativi impoverendo l’It service delivery. Situazione che non risulta però più sostenibile rispetto al contesto economico in cui operano le aziende ‘servite’ dall’It in cerca di maggior flessibilità, dinamicità e velocità di risposta, senza però perdere in qualità e sicurezza. Ecco perché, secondo Gartner, sta crescendo ed evolvendo la metodologia DevOps che mira a migliorare l’It service management.

Sondaggio Gartner: infrastrutture IT spesso very complex, Analizzando un po’ più in dettaglio le ‘cause’ di tale complessità, Gartner evidenzia come le problematiche maggiori siano da rilevare nell’ambito dei processi It, da un lato, e nel proliferare di tool tecnologici, dall’altro. Situazioni che rendono difficoltoso sia il rilascio sia il mantenimento di servizi It efficienti (ed efficaci sul piano del business).

E dato che uno dei servizi It più impattante sul piano del business è quello applicativo, Gartner ha voluto indagare per capire meglio quali possano essere le ‘disfunzioni’ all’interno della famiglia It che si riflettono poi sull’azienda. Una delle aree maggiormente critiche risulta essere la ‘relazione’ tra l’application development e le It operation. Ben il 47% degli intervistati durante la conferenza di Las Vegas ha dichiarato che i rapporti tra questi differenti team è del tutto ‘un-collaborative’. Le lamentele più comuni delle operation, scrive Haight nel suo report, derivano dal fatto che spesso queste vengono ‘lasciate fuori’ dalle decisioni sia inerenti le più importanti architetture It sia quelle di business, con il risultato di un minor controllo sui service level (compresi i costi).

I principali ‘conflitti’, causa della ‘disfunzionale’ relazione tra development e operation, derivano, infatti, dalle persone, poco abituate a un approccio collaborativo anche, e soprattutto, a causa delle strutture ‘a silos’ dei data center.

La risposta, come dicevamo, può essere l’approccio DevOps. Trattandosi però di un approccio nuovo, ancora poco conosciuto e per il quale non esistono ancora metodologie e best practice consolidate (il 44% degli intervistati dichiara di essere ancora ‘fermo’ nel tentativo di capire di cosa si tratti; solo il 20% ha già iniziato un qualche tipo di percorso), gli impedimenti sono ancora di natura culturale (resistenza al cambiamento delle persone e dei processi – 47%) 🡪 importante anche l’aspetto psicologico che deve essere propenso al cambiamento.

A confermare l’interesse verso questa nuova metodologia i dati relativi ai vantaggi percepiti sia da parte dei development team sia da parte delle It operation: release management (36%) e change management (35%) sono i principali processi che, secondo gli intervistati, beneficerebbero maggiormente dall’adozione di pratiche e iniziative di DevOps, con interessanti riscontri anche sul piano del performance management (12%) e del capacity planning (12%) (vedi figura 3).

I team si affidano all'automatizzazione per velocizzare processi manuali che sono da sempre lenti. Le tecnologie e gli strumenti impiegati aiutano i team a far funzionare ed evolvere le applicazioni in modo rapido e affidabile. Questi strumenti aiutano i tecnici anche a portare a termine attività che normalmente avrebbero richiesto l'attenzione di altre unità aziendali, ad esempio la distribuzione di codice o il provisioning dell'infrastruttura, aumentando ulteriormente l'efficienza.

Perché l’approccio DevOps è importante

Il software e Internet sono fattori che hanno cambiato il mondo e la sua economia, a partire da shopping, intrattenimento e settore bancario. Il software non è più solo un supporto per il business; è diventato un componente integrante di qualsiasi organizzazione. Le aziende interagiscono con i loro clienti tramite prodotti software sotto forma di applicazioni o servizi online che è possibile utilizzare su molti dispositivi differenti. Impiegano altri prodotti software per migliorare l'efficienza operativa apportando trasformazioni a qualsiasi anello della catena di valore, ad esempio logistica, comunicazioni e operazioni. Le aziende di oggi, quindi, devono trasformare il modo in cui creano e distribuiscono il loro software, come le industrie che creano beni materiali hanno trasformato nel corso del XX secolo il modo in cui progettavano, creavano e distribuivano i loro prodotti attraverso l'automazione industriale.

Prassi principali:

* Cultura aziendale
* Una prassi fondamentale è aumentare la frequenza degli aggiornamenti e diminuirne le dimensioni. Questo è uno dei modi con cui un'azienda può innovare più rapidamente a beneficio dei propri clienti. Questi aggiornamenti solitamente hanno una natura maggiormente incrementale rispetto agli aggiornamenti saltuari previsti dalle prassi di rilascio di software tradizionali. Quando la frequenza degli aggiornamenti è maggiore e le dimensioni minori è più facile tenere sotto controllo il rischio. I bug vengono risolti più rapidamente perché è più facile individuare quale implementazione ha provocato il problema. Anche se la cadenza e le dimensioni degli aggiornamenti variano, le aziende che adottano un modello DevOps distribuiscono aggiornamenti più di frequente rispetto a quelle con modelli di sviluppo di software tradizionali.
* Utilizzo di microservizi: scomposizione dell’applicazione in tanti servizi indipendenti tra loro e con una sola funzione

L'utilizzo di microservizi e la maggiore frequenza delle release aumentano tuttavia il numero di distribuzioni, il che può presentare problemi dal punto di vista operativo. Le prassi DevOps come l'integrazione continua e la distribuzione continua risolvono questi problemi e consentono alle organizzazioni di velocizzare le distribuzioni in modo sicuro e affidabile. Le prassi di automatizzazione dell'infrastruttura, come ad esempio l'infrastruttura come codice e la gestione della configurazione, aiutano a mantenere elasticità e tempi di risposta ottimali delle risorse di organizzazione. Monitoraggio e registrazione di log, inoltre, aiutano i tecnici a tenere sotto controllo le prestazioni di applicazioni e infrastruttura, per rispondere in modo tempestivo ai problemi.

Continuos integration:

L'integrazione continua è un metodo di sviluppo software in cui gli sviluppatori aggiungono regolarmente modifiche al codice in un repository centralizzato, quindi la creazione di build e i test vengono eseguiti automaticamente. Gli obiettivi principali dell'integrazione continua sono individuare e risolvere i bug con maggiore tempestività, migliorare la qualità del software e ridurre il tempo richiesto per convalidare e pubblicare nuovi aggiornamenti.

Continuos delivery:

La distribuzione continua è un metodo di sviluppo software in cui le modifiche al codice vengono applicate a una build, testate e preparate per il rilascio in produzione in modo automatico. Estende l'integrazione continua distribuendo tutte le modifiche al codice all'ambiente di testing e/o di produzione dopo la fase di creazione di build. Se la distribuzione continua viene implementata correttamente, gli sviluppatori hanno sempre a disposizione una build temporanea pronta per la distribuzione che ha già passato un processo di testing standardizzato.

Infrastructure as a code: (Vagrant)

L'infrastruttura come codice è una prassi secondo cui provisioning e gestione dell'infrastruttura avvengono tramite metodologie di sviluppo di software e codice quali il controllo di versione e l'integrazione continua. Il modello di gestione del cloud tramite API permette a sviluppatori e amministratori di sistema di interagire con l'infrastruttura in modo programmatico e su larga scala, piuttosto che tramite l'impostazione e la configurazione manuale delle risorse. Di conseguenza, i tecnici possono interfacciarsi con l'infrastruttura mediante strumenti basati sul codice e secondo le stesse modalità previste per il codice di un'applicazione. Poiché possono essere definiti tramite codice, l'infrastruttura e i server possono essere distribuiti con la massima rapidità tramite modelli standardizzati, aggiornati con patch e versioni più recenti o duplicati in modo iterabile.

Gestione della configurazione

Sviluppatori e amministratori di sistema utilizzano codice per automatizzare sistemi operativi e mantenere in hosting configurazione, attività operative e altri processi. L'utilizzo di codice rende le modifiche alla configurazione iterabili e standardizzate. Solleva sviluppatori e amministratori di sistema dal compito di configurare manualmente sistemi operativi, applicazioni di sistema o software dei server.

Monitoraggio e registrazione di log

Le aziende tengono sotto controllo i parametri e i log per scoprire in che modo le prestazioni di applicazione e infrastruttura influiscano sull'esperienza dell'utente finale. Per capire l'impatto di modifiche o aggiornamenti sugli utenti finali, le aziende devono acquisire dati e log generati da applicazioni e infrastruttura, suddividerli in categorie e analizzarli, esaminando le possibili cause primarie dei problemi o le modifiche impreviste. Il monitoraggio attivo diventa ancora più importante quando la disponibilità del servizio non deve presentare interruzioni e a mano a mano che aumenta la frequenza degli aggiornamenti di applicazione e infrastruttura. Anche la creazione di allarmi o l'analisi dei dati in tempo reale sono attività che aiutano le organizzazioni a monitorare proattivamente i loro servizi.

DevOps permette alle organizzazioni di consegnare software migliori, più velocemente e consentendo un’innovazione continua: la ricerca “State of DevOps 2017” di Puppet Labs ha indicato che i team che utilizzano pratiche DevOps rilasciano aggiornamenti con una frequenza 46 volte superiore e con tempi di risposta 440 volte più brevi.

Il metodo DevOps nasce dall’incontro tra development e operations. Questo nuovo metodo, in sintesi, spinge alla collaborazione, alla comunicazione e all’integrazione del lavoro tra gli sviluppatori di software e i sistemisti, fornendo all’organizzazione l’abilità di sviluppare applicazioni e servizi con la massima agilità e ad una maggiore velocità. La maggiore agilità consente alle aziende di offrire servizi migliori ai clienti e offre una maggiore competitività sul mercato. (buona introduzione). Per far sì che i team dedicati a sviluppo e produzione non agiscano più separatamente, è necessaria un'evoluzione di mentalità e cultura aziendali. In alcune organizzazioni i due gruppi potranno addirittura essere fusi in un'unica unità. Le aziende che adottano un modello DevOps, indipendentemente dalla loro struttura organizzativa, dispongono di team che annoverano tra le loro responsabilità il controllo dell'intero ciclo di vita di sviluppo e infrastruttura. Prima della nascita della metodologia di lavoro DevOps, gli sviluppatori di software (dev) e i sistemisti (ops) erano separati, anche fisicamente. Tra i due team non vi era comunicazione diretta, il lavoro era a silo, e le due metà comunicavano esclusivamente per ticket (normalmente inviati dai dev agli ops). Con l’approccio DevOps, il modo di lavorare cambia: questo prevede, infatti, che fin dall’inizio del progetto si formino due team congiunti, dev + ops. Quando, quindi, il project manager dà il via al progetto, mette su un team di dev che includa almeno un sistemista, il quale ha il compito di gestire le infrastrutture lato cliente (vale a dire, supportare i dev nel gestire i sistemisti dei clienti stessi) e lato interno. Attraverso questa sinergia, non si ha più un sistema basato sui ticket, ma un sistema agile, basato sulla comunicazione. Favorendo la comunicazione, si minimizza il rischio di errori e soprattutto il rischio che emergano errori banali e facilmente evitabili.

Velocità: Processi più agili per velocizzare l’innovazione, migliorare l’adattamento ai cambiamenti dettati dal mercato e aumentare l'efficienza.

Automazione: Aumenta l’efficienza, grazie alla possibilità di automatizzare una serie di processi ripetitivi, e migliora la qualità del programmi grazie all’utilizzo di software in grado di fare test automatici.

Distribuzione rapida: Aumenta la frequenza delle nuove release per facilitare innovazione e miglioramento dei prodotti. La maggiore rapidità nel rilascio di nuove funzioni e correzioni di bug aumenterà il vantaggio sulla concorrenza. Inoltre, integrazione e distribuzione continua automatizzano le procedure di rilascio del software, dalla build alla distribuzione.

Affidabilità: Verifica che gli aggiornamenti delle app e le modifiche dell'infrastruttura siano sempre conformi agli standard di qualità per garantire affidabilità e produttività. Con l'integrazione e la distribuzione continua potrai verificare la sicurezza di ogni modifica. Col monitoraggio e la registrazione di log potrai mantenere sotto controllo le prestazioni in tempo reale.

Scalabilità: Gestisci l'infrastruttura e i processi di sviluppo su qualsiasi scala. Automazione e coerenza consentono di gestire sistemi complessi e soggetti a variazioni in modo efficiente e contenendo i rischi. Ad esempio, l'utilizzo di infrastrutture come codice permette di gestire ambienti di sviluppo, testing e produzione in modo iterabile e più efficiente.

A partire dal 2009 si è assistito a una convergenza verso quello che in gergo viene definito DevOps, una metodologia di sviluppo del software che trae i suoi fondamenti dall’agile methodology e ha come obiettivo il superamento del tradizionale modello a cascata. Nel mondo del software sta accadendo quello che per certi versi è avvenuto nel mondo del manufacturing, quando Toyota introdusse il concetto di Lean Production definendo le basi di un nuovo sistema di produzione improntato a una maggiore rapidità di esecuzione e standard di qualità.