

4 insidie nello sviluppo di applicazioni low-code:

Ripensa il tuo approccio ed evita gli errori comuni.

1

Perché è importante un approccio low-code allo sviluppo delle applicazioni.

2

Insidia n. 1: il team di sviluppo è troppo grande e specializzato.

4

Insidia n. 2: non stai analizzando abbastanza a fondo il problema.

5

Insidia n. 3: i feedback sono lenti, insufficienti e distorti.

6

Insidia 4: il successo si misura solo al momento del lancio.

7

Come ottimizzare lo sviluppo low-code.

8

Informazioni su Appian

I team IT di tutti i settori stanno abbracciando il low-code come nuovo modo di costruire applicazioni.

Non è solo più veloce (produce applicazioni potenti a 10 volte o più la velocità dello sviluppo tradizionale). Se fatto bene, il low-code permette di fornire funzionalità migliori, contare su team di sviluppo più piccoli e versatili e ottimizzare la collaborazione con gli utenti.

Sfortunatamente, molti team IT stanno solo iniziando a scoprire il potenziale del low-code. Perché? Perché stanno eseguendo progetti low-code con approcci high-code.

Per sfruttare appieno il low-code, non si può semplicemente sostituire Java con una piattaforma low-code ed eseguire tutto come si è sempre fatto. Occorre rinunciare ad approcci di sviluppo obsoleti e appesantiti e adottare invece alcuni approcci chiave per la progettazione di team e prodotti.

Scopri le quattro insidie più comuni dello sviluppo di applicazioni low-code e come puoi combatterle con un moderno approccio low-code.



Per scoprire il pieno potenziale del low-code, sostituisci le prassi di sviluppo datate con alcuni approcci chiave più agili.

Insidia n. 1: il team di sviluppo è troppo grande e specializzato.



I team di sviluppo delle applicazioni tradizionali sono grandi, specializzati e pieni di inter- e intra-dipendenze eccessive.

I migliori team low-code sono fatti per essere agili e minimizzare le dipendenze.

Sono piccoli e hanno tutte competenze necessarie per passare dall'idea all'applicazione. I team rimangono invariati di progetto in progetto in modo da non dover dedicare altro tempo per imparare a collaborare con nuove persone.

I team low-code sono progettati per essere collaborativi ed egualitari, pieni di persone che possono essere descritte come "sviluppatore e" (ad esempio, sviluppatore e tester o sviluppatore e Scrum Master). Tutti, dai team leader agli architetti, svilupperanno funzionalità. E ogni sviluppatore deve avere almeno un'ulteriore area di specializzazione:— design UX, test, architettura, ecc.

Nel creare la tua squadra, assicurati di avere persone competenti in queste sei aree di specializzazione:

Product ownership... Questo è l'unico ruolo che non sviluppa funzionalità. Il product owner deve esprimere ciò che è necessario dal punto di vista del business, dare priorità al valore e garantire che i requisiti siano visibili, trasparenti e chiari per tutti.

UX design. Almeno un ruolo nel team deve coprire l'UX design. Questa persona può assicurare che vengano seguiti buoni principi di design e creare progetti di soluzioni per decisioni importanti sul fronte del design.

Progettazione e architettura tecnica. Questo ruolo assicura che il team costruisca l'applicazione nel modo giusto stabilendo l'architettura della soluzione.

Analista di business. Questo ruolo esamina i requisiti di business e le esigenze degli utenti, facendo domande approfondite e considerando ogni scenario per trasformare processi complessi in qualcosa che possa essere codificato in una soluzione.

Testing. Oltre allo sviluppo di funzionalità, questo ruolo si concentra sui test esplorativi per trovare aree di rottura non rilevate dagli script dei test. Chi lavora in questa posizione deve anche conoscere gli strumenti per i test automatici e i test di performance e di carico.

Team lead o Scrum Master. Questo ruolo consente al team di costruire rapidamente, coordinando il lavoro di sviluppo, rimuovendo gli impedimenti e incoraggiando pratiche agile.

Il low-code aiuta a mantenere le cose semplici, compresi i team delle applicazioni.

Poiché il low-code è più facile da imparare rispetto all'high-code, gli specialisti in campi come UX, etnografia, testing e analisi di business possono seguire una formazione incrociata come sviluppatori low-code. Questi nuovi "specialisti generalizzati" hanno competenze sovrapposte che permettono di costruire i team di sviluppo in modo diverso e sfruttare appieno il low-code. I team ottimizzati low-code hanno costi inferiori sia in termini di costo di lavoro (meno persone, utilizzate appieno) che di prodotto (i team veramente interfunzionali e collaborativi hanno molte più probabilità di costruire il prodotto giusto). I team low-code forniscono un time to value più veloce perché la loro composizione versatile rimuove le strozzature interne e riduce le dipendenze esterne (ad esempio, non c'è bisogno di aspettare che si liberi uno specialista di database se il team comprende già un esperto di database e uno sviluppatore low-code).



Gli approcci di sviluppo tradizionali fanno affidamento su analisti di business che fungono da traduttori tra gli utenti di business e i team tecnologici. I due domini sono separati l'uno dall'altro, il che significa che le persone con il problema sono separate da quelle più equipaggiate per risolverlo: gli sviluppatori. Gli sviluppatori che cercano di comprendere appieno il problema sono ostacolati da canali di comunicazione a bassa fedeltà (ad esempio le e-mail), traduttori imperfetti, latenza, eccessivo affidamento alla documentazione e, cosa più dannosa di tutte, nessun accesso agli utenti finali.

Invece di programmatori con un decennio di esperienza in Java, cerca sviluppatori che siano più interessati a risolvere problemi che a scrivere codice. Questi collaboratori possono avere una o due specializzazioni tecniche così come una conoscenza generale dello sviluppo del software e dei settori di business in cui lavorano.

Nella ricerca delle persone giuste, ti consigliamo di promuovere una cultura delle domande aperte. Invece di chiedere “La conformità è un problema?”, chiedi “Cosa succede se la conformità non viene rispettata?” Non prendere mai i requisiti per buoni senza metterli in discussione. Incoraggia gli sviluppatori ad andare a fondo, chiedi sempre “Perché?” e troverai soluzioni non lineari.

Questo ha inizio con l'avvio del progetto— spesso chiamato Sprint 0— con sessioni collaborative che includono IT, sponsor aziendali e utenti finali. Lavorate insieme per definire gli obiettivi, esplorate ciò che l'applicazione deve fare per raggiungere quegli obiettivi e tracciate un piano per consegnare un prodotto di valore. L'intento non è quello di definire ogni dettaglio del progetto, ma di pianificare abbastanza bene da iniziare con il piede giusto.

Parla con gli utenti aziendali di queste cinque aree chiave:

Visione

Cosa vuoi ottenere con l'applicazione?

Interlocutori

Quali utenti finali e altri interlocutori ne beneficeranno?

Esigenze

Qual è il problema o il bisogno che l'applicazione deve affrontare? Quali sono i punti critici, le strozzature e i vantaggi auspicati?

Applicazione

Quali sono le funzioni e le caratteristiche indispensabili? Perché sono “must-have”? Esiste una catena causale che collega la funzione a un obiettivo aziendale misurabile?

Obiettivi di business

Quali sono gli obiettivi aziendali misurabili, in ordine di priorità?

Ricorda anche di valutare lo stato attuale dei tuoi utenti. Poiché la tua nuova applicazione cambierà il loro modo di lavorare, dovrai capire cosa vogliono fare i tuoi interlocutori e cosa li sta ostacolando. Il modo migliore per farlo è quello di “supervisionare l'ambiente” con utenti reali, guardandoli lavorare nella loro vita quotidiana. Se questo non è possibile, chiedi agli utenti di descrivere i propri processi quotidiani, concentrandosi sul perché eseguono ogni attività e quali sono le più grandi difficoltà che incontrano.

Insidia n. 3: i feedback sono lenti, insufficienti e distorti.



Il low-code rimuove gli ostacoli che rallentano i progetti high-code. Con le practice di sviluppo tradizionali, i cicli di feedback sono come un traghetto che viaggia tra due rive. Gli utenti aziendali sono da una parte e gli sviluppatori che costruiscono l'applicazione sono dall'altra. Si traghettano le informazioni avanti e indietro lentamente, in un processo ad alta latenza che rischia molte distorsioni.

Le migliori soluzioni emergono quando gli utenti aziendali e gli sviluppatori creano applicazioni insieme. Invece di trasportare il feedback avanti e indietro, il low-code costruisce un ponte tra sviluppatori e utenti aziendali, facilitando la collaborazione e incoraggiando input chiari e frequenti; in questo modo potrai testare le ipotesi in corso d'opera mentre ti dirigi verso la soluzione giusta.

L'aumento della collaborazione e lo sviluppo iterativo aiutano anche a ottenere un buy-in precoce da parte degli interlocutori. Piuttosto di uno sviluppo isolato che poi genera tutto il valore alla fine del progetto, gli utenti finali possono approvare e influenzare l'applicazione proprio mentre prende forma davanti ai loro occhi. I prototipi e le versioni in corso di sviluppo danno agli utenti qualcosa di concreto su cui esprimersi, accelerando il processo di design e assicurando che il tuo team di sviluppo abbia una chiara comprensione di ciò che serve.



Invece di traghettare lentamente i feedback avanti e indietro, il low-code costruisce ponti tra gli sviluppatori e gli utenti aziendali.

L'implementazione è una pietra miliare importante. Ma se la tua applicazione non viene usata o se gli utenti non sono soddisfatti, non porterà mai valore.

Questo ha inizio con lo Sprint 0 nel momento in cui ti occupi di comprendere appieno le esigenze e le problematiche dei tuoi utenti, creando la base per la tua soluzione.

Continua a concentrarti sui tuoi utenti finali durante l'intero ciclo di vita dello sviluppo. Individua alcuni interlocutori chiave molto rispettati all'interno del gruppo di utenti e coinvolgili quanto prima e quanto più spesso possibile per coltivare un buon rapporto. Ecco alcune idee per coinvolgere i tuoi evangelisti nella soluzione:

- Invitali a partecipare al processo di progettazione affinché diano il proprio contributo e facciano domande. Poi chiedi loro di riferire all'intero gruppo di utenti ciò che hanno visto.
- Consegna le chiavi della demo in modo che possano eseguire una dimostrazione per l'intero gruppo di utenti.
- Raccogli i feedback da loro e da altri utenti.

Crea opportunità affinché il team IT si faccia da parte lasciando spazio agli utenti. L'obiettivo è che l'applicazione diventi tanto loro quanto vostra.

Il low-code aiuta a costruire applicazioni veloci e da subito corrette. Per vedere il pieno potenziale del low-code, vai oltre gli approcci di sviluppo tradizionali e segui questi consigli:

- Semplifica lo sviluppo con un team agile e compatto.
- Definisci il problema giusto con una profonda comprensione dei tuoi utenti.
- Facilita la collaborazione con gli utenti aziendali.
- Proponi applicazioni di successo che vengano adottate e amate.

Non importa se hai appena iniziato con il low-code o se stai diffondendo il low-code nella tua organizzazione. L'adozione di alcune best practice agile può trasformare tempistiche lunghe e applicazioni appena accettabili in uno sviluppo rapido dai risultati eccezionali.

Per ulteriori best practice, strumenti, modelli e altre risorse per ottimizzare il tuo lavoro di sviluppo di applicazioni low-code consulta [Appian Delivery Methodology](#).



L'adozione di alcune best practice agile può trasformare tempistiche lunghe e applicazioni appena accettabili in uno sviluppo rapido dai risultati eccezionali.

Appian permette alle organizzazioni di creare rapidamente applicazioni e workflow con una piattaforma di automazione low-code. Combinando persone, tecnologie e dati in un unico workflow, Appian aiuta le aziende a massimizzare le risorse e ottimizzare i risultati. Molte delle maggiori organizzazioni al mondo sfruttano le applicazioni Appian per migliorare la customer experience, raggiungere l'eccellenza operativa e semplificare la gestione del rischio e la compliance a livello globale.

I membri del team Appian Customer Success sono i principali esperti di Appian. La nostra cultura consiste nel fare ciò che è giusto per i nostri clienti. Il nostro obiettivo è fornire rapidamente i migliori risultati di business per i clienti utilizzando la nostra piattaforma di automazione low-code.

Fai esperienza pratica con [Appian Community Edition](#) o contatta success@appian.com per scoprire come Appian Customer Success può aiutarti con i tuoi progetti low-code.



Ciò che ci ha sorpreso di più è la velocità con cui siamo in grado di sviluppare un'applicazione e implementarla.

Ryder

Per saperne di più vai su it.appian.com
Contattaci all'indirizzo info.it@appian.com



Appian aiuta le organizzazioni a costruire applicazioni e workflow rapidamente, con una piattaforma di automazione low-code. Combinando persone, tecnologie e dati in un unico flusso di lavoro, Appian può aiutare le aziende a massimizzare le risorse e migliorare i risultati di business. Molte delle più grandi organizzazioni al mondo utilizzano le applicazioni Appian per migliorare l'esperienza del cliente, raggiungere l'eccellenza operativa e semplificare la gestione del rischio globale e la compliance. Per ulteriori informazioni, visita il sito it.appian.com

