Malorean Corp.

Un romanzo in cui riconoscersi

Una panoramica per capire

Sviluppatori, IA e piccola impresa : una storia (quasi) vera di trasformazione

Ciò che vivono davvero i team di un SaaS ERP con l'IA oggi



LAURENT DESÉCHALLIERS

Malorean Corp

> *Fino a poco tempo fa, trasformare un'idea in un prototipo poteva richiedere settimane. Oggi, con l'aiuto dell'intelligenza artificiale, una persona può riuscirci in pochi giorni. È un cambiamento straordinario. Ma la vera trasformazione potrebbe avvenire altrove: nel modo in cui lavoriamo insieme.*

Laurent Deséchalliers

Nota sulla traduzione

Questa edizione è una traduzione automatica, prodotta da un'intelligenza artificiale a partire dal testo francese.

Non è stata riletta da un traduttore umano. Le idee, la struttura e gli esempi sono fedeli all'originale; alcune formulazioni, invece, possono suonare goffe, e qualche passaggio conserverà la traccia della lingua di partenza.

È il prezzo da pagare perché quest'opera esista in più lingue anziché soltanto in francese. In caso di dubbio su una formulazione, l'edizione francese fa fede.

Introduzione

Ciò che più mi colpisce dell'IA non è la sua capacità di scrivere codice. È la velocità con cui rende possibile qualcosa che, fino a poco tempo fa, avrebbe richiesto giorni o settimane di lavoro.

Un'idea può diventare un prototipo prima ancora che si sia conclusa la discussione sulla sua fattibilità. Uno sviluppatore può esplorare una strada senza aspettare che sia disponibile un intero team. A volte anche chi non sa programmare può costruire una prima versione di ciò che immagina. Questo non significa che un prototipo sia un prodotto affidabile, né che la complessità sia scomparsa. Ma la sola possibilità di provare cambia già il nostro rapporto con ciò che sembrava possibile.

E le nostre aziende non sono state costruite per un mondo in cui realizzare una prima soluzione può diventare così rapido.

Si sono organizzate intorno alle competenze, al tempo disponibile, ai passaggi di consegne e alle scelte necessarie per far avanzare il lavoro. Sono meccanismi che hanno una loro ragion d'essere. Eppure, quando cambiano le capacità delle persone, cominciano a emergere nuove domande. Dobbiamo ancora lavorare nello stesso ordine? Aspettare le stesse approvazioni? Riservare le stesse attività alle stesse professioni?

Queste domande non riguardano soltanto gli sviluppatori. Anche i Product Manager, i designer, i commerciali, i responsabili dei team e molti altri scoprono nuovi modi di lavorare. Alcuni possono ora svolgere direttamente una parte di ciò che prima chiedevano ai colleghi. Altri vedono cambiare la natura di una parte della propria attività. Quella che sembrava una

questione di strumenti comincia a toccare il modo in cui collaboriamo e prendiamo decisioni.

Sarebbe facile annunciare una rivoluzione di cui conosciamo già l'esito. Non credo che siamo a quel punto.

Alcune aziende sperimentano nuove distribuzioni del lavoro e delle responsabilità. Altre procedono con maggiore prudenza, talvolta per buone ragioni. Certe pratiche sembrano promettenti, ma ci manca ancora il distacco necessario per sapere quali resisteranno ai vincoli della vita quotidiana. E una soluzione adatta a un piccolo team non è automaticamente pertinente per un'azienda più grande, più regolamentata o con una storia più lunga.

Non ho quindi voluto scrivere un manuale su come organizzarsi correttamente con l'IA. Questo libro è un tentativo di capire che cosa stia cambiando davvero: ciò che gli strumenti rendono possibile, come reagiscono le aziende, che cosa guadagnano, che cosa rischiano di perdere e quali questioni dovranno affrontare.

Per farlo, ho scelto due percorsi che si incrociano.

Il primo è quello dell'osservazione. Esamineremo aziende reali, pratiche rese pubbliche, studi ed esperienze. Cercherò di distinguere ciò che è accertato da ciò che rimane un'ipotesi, senza confondere un annuncio accattivante con un risultato dimostrato.

Il secondo è quello di una storia. Seguiremo Malorean Corp., un'azienda immaginaria i cui dipendenti scopriranno gradualmente come queste nuove possibilità cambiano il loro lavoro. Non hanno un piano per reinventare l'organizzazione. Hanno clienti, abitudini, urgenze, convinzioni e, a volte, disaccordi. La finzione non dimostra nulla; ci permette di osservare queste trasformazioni dal punto di vista di chi le vive.

Entrambe le prospettive sono necessarie. Gli esempi reali aiutano a non scambiare le nostre intuizioni per fatti. La storia ci ricorda che i cambiamenti organizzativi non si svolgono soltanto negli organigrammi: riguardano la fiducia, le competenze, le responsabilità e il posto di ciascuno.

In fondo, una domanda attraverserà le pagine che seguono: **che cosa succede in un'azienda quando le persone che ne fanno parte scoprono di poter fare molto più di quanto la loro professione permettesse loro fino a quel momento?**

L'autore

Lavoro nel settore del software da diversi decenni e oggi sono Product Manager. Il mio percorso mi ha portato dallo sviluppo e dall'architettura software all'organizzazione dei team, poi al prodotto, alle attività operative e al marketing che lo riguardano.

Questa esperienza mi offre un punto di osservazione privilegiato sull'evoluzione dell'intelligenza artificiale e sui suoi effetti sul nostro lavoro. Li vedo ogni giorno nella mia attività professionale, ma anche attraverso gli scambi con i colleghi Product Manager. Condividiamo esperienze, scoperte e interrogativi. Queste conversazioni mi portano spesso a guardare oltre gli strumenti: che cosa cambia davvero nelle nostre professioni, nelle nostre decisioni e nel modo in cui lavoriamo insieme?

Con il susseguirsi di osservazioni e discussioni, le domande si sono moltiplicate. Ho sentito il bisogno di prendere le distanze, confrontare le nostre impressioni con ciò che accade in altre aziende e condividere questa riflessione. È così che è nato questo libro.

La scrittura accompagna da tempo il mio percorso. Ho collaborato con la stampa specializzata e lavorato come revisore tecnico per M2 Édition ed Eyrolles. Oggi scrivo su malorean.net di Product Management, IA, metodi di lavoro e sistemi informativi.

Collaboro anche con *Proposition Marketing*, una rivista indipendente dedicata al marketing strategico e operativo. Scrivere per lettori che si occupano di marketing e direzione aziendale mi spinge a rendere acces-

sibili i temi di prodotto, senza dare per scontato che tutti condividano lo stesso linguaggio.

Laurent Deséchalliers

- laurent@malorean.net
- https://malorean.net

Dello stesso autore



Le Contexte — Come costruire un contesto che permetta all'IA di capire ciò che conta, collegare le proprie conoscenze e mettere a frutto ciò che si impara al servizio dei propri obiettivi.
Uscita: ottobre 2026.



NO IA — *Non delegare la tua comprensione.* Metodi senza IA per coltivare la nostra memoria, approfondire la nostra riflessione ed esercitare il nostro spirito critico. Pratiche per sviluppare le nostre capacità e sfruttare meglio l'IA senza delegarle la nostra comprensione.
Uscita prevista: fine 2026.

Maggiori informazioni su https://malorean.net.

Ringraziamenti

Desidero innanzitutto ringraziare i Product Manager con cui mi confronto regolarmente. Mettiamo a paragone i nostri punti di vista sull'evoluzione del Product Management, delle professioni legate allo sviluppo software e, più in generale, sugli effetti dell'intelligenza artificiale sul lavoro e sull'organizzazione delle aziende. Queste conversazioni, a volte animate, sono una fonte continua di domande, ripensamenti e nuove idee. Alcune di queste riflessioni si ritrovano nelle pagine di questo libro. Grazie per la vostra curiosità, per le esperienze che condividete e per la voglia di discutere.

E poi voglio ringraziare Christine, mia moglie. Alla fine del mio primo libro, *Le Contexte*, le avevo promesso una pausa. Invece ho cominciato quasi subito a scrivere questo. Per fortuna, questa seconda avventura ha occupato meno delle nostre serate e dei nostri fine settimana rispetto alla precedente. Grazie, Christine, per la tua pazienza e per il tuo sostegno, anche quando le mie riflessioni finiscono per diventare un altro libro.

Nota sugli esempi

Le riflessioni e gli esempi di questo libro nascono da oltre trent'anni di esperienza professionale nel settore informatico. Provengono da situazioni incontrate in progetti e ambienti di lavoro diversi, e da numerosi scambi con product manager, CTO, engineering manager e altri professionisti di varie aziende. Riguardano quindi più organizzazioni e diverse fasi del mio percorso.

Per renderle utili al libro, ho modificato, generalizzato o ricomposto queste situazioni. I casi d'uso, i dati numerici e le metriche illustrano problemi ricorrenti e ciò che ne ho imparato. Sono costruzioni: nessuna descrive un cliente, un prodotto o un'organizzazione reali, né riflette i dati, le pratiche o le posizioni del mio datore di lavoro.

Le analisi, i metodi e le opinioni espressi in questo libro sono personali e riflettono esclusivamente il mio punto di vista.

Questo libro è stato scritto con l'IA?

Sì. Interamente in collaborazione con l'intelligenza artificiale.

So che oggi questa domanda ha un'importanza particolare. Ci si chiede se un articolo, un romanzo o un libro sia stato scritto con l'IA. C'è persino chi sottopone i testi a programmi di rilevamento, come se fosse indispensabile distinguere ciò che proviene da una persona da ciò che proviene da una macchina.

Preferisco essere trasparente: l'IA ha partecipato a ogni fase di questo libro, dalla ricerca documentale alla stesura, passando per l'organizzazione delle idee, la verifica delle fonti e l'individuazione dei punti ciechi. Ma questa collaborazione richiede una spiegazione, perché da sola non dice chi abbia concepito il libro.

Per molto tempo, per svolgere un lavoro di questa portata era possibile circondarsi di persone incaricate di cercare documenti, mettere a confronto idee, rileggere, strutturare o impaginare. Oggi l'IA mi dà accesso a una parte di queste capacità. Esegue rapidamente i compiti, propone formulazioni, segnala incoerenze e a volte pone domande a cui non avevo pensato. Mi permette di esplorare molto più di quanto potrei fare da solo nel tempo di cui dispongo.

Per scrivere questo libro ho utilizzato un motore di Contesto basato sulle tecnologie di ChatGPT, con Notion per conservare le informazioni nel tempo. Fonti, idee, decisioni e versioni di lavoro possono così essere mantenute e recuperate da una conversazione all'altra. Ho già presentato

questo approccio nel mio primo libro, *Le Contexte*. Lì spiego come un sistema del genere permetta all'IA di diventare una sorta di supersegretario personale: non si limita più a rispondere a una domanda, ma tiene il filo dei progetti e mi aiuta a portarli avanti nel tempo.

Esiste però una distinzione essenziale: **essere capaci di eseguire non significa assumersi la responsabilità di capire e scegliere.** Le domande che guidano questo libro, l'esperienza professionale che le alimenta, le interpretazioni, le scelte e ciò che alla fine decido di sostenere o scartare sono miei. L'IA può contraddirmi, mettermi alla prova e aiutarmi a formulare meglio un pensiero. Non sostituisce né il mio giudizio né la mia responsabilità di autore.

Distinguo inoltre questo libro da opere in cui la scrittura stessa è una parte fondamentale della creazione. Quando leggiamo Shakespeare nella traduzione di François-Victor Hugo, oppure Cervantes o Cizia Zykë nel testo originale, la voce, il ritmo e lo stile contano quanto ciò che viene raccontato. In un'opera tradotta, anche le scelte del traduttore fanno parte dell'esperienza. Il ruolo che desideriamo attribuire all'IA in questo genere di letteratura solleva altre domande, che non pretendo di risolvere qui.

Questo libro contiene una parte di finzione, ma la sua ambizione principale resta comprendere una trasformazione tecnica e organizzativa. La forma deve essere gradevole e chiara; è soprattutto al servizio delle idee, dei fatti e delle domande che desidero condividere.

L'IA ha reso possibile questo progetto su una scala che probabilmente non avrei potuto raggiungere con le mie sole risorse, accanto alla mia attività professionale. Mi permette di sviluppare progetti personali, come questo libro, senza rinunciare all'impegno di riflessione che richiedono.

E in fondo c'è una certa coerenza nell'aver scritto un libro sulle trasformazioni provocate dall'IA lavorando proprio in questo modo.

Laurent Deséchalliers

Indice

Nota sulla traduzione 3

Introduzione 4

L'autore 7

Dello stesso autore 9

Ringraziamenti 10

Nota sugli esempi 11

Questo libro è stato scritto con l'IA? 12

Malorean Corp. 17

1 Malorean Corp. — Una giornata fuori dal comune 18
2 L'IA ha moltiplicato la capacità di sviluppo software 24
3 Malorean Corp. — Il giorno dopo 33
4 Stanno emergendo nuovi modelli organizzativi 46
5 Malorean Corp. — Ciò che possiamo fare e ciò che dobbiamo fare 55
6 I confini tra i ruoli si sfumano, le responsabilità restano 68
7 Malorean Corp. — Ciò che Alice sapeva senza averlo mai scritto 78
8 Più autonomia richiede regole più chiare, non meno regole 94
9 Malorean Corp. — Due decisioni ragionevoli 109
10 L'IA farà scomparire i livelli intermedi delle aziende? 123
11 Malorean Corp. — Il tempo ritrovato 138
12 Conclusione 155

Malorean Corp.

1 – Malorean Corp. – Una giornata fuori dal comune

La sede di Malorean Corp. si trovava a Le Havre. Ma dopo la pandemia, quell'indirizzo diceva ben poco su dove si svolgesse davvero il lavoro. I dipendenti erano sparsi in tutta la Francia. Si incontravano davanti a una videocamera, discutevano condividendo lo schermo e, il più delle volte, nessuno aveva bisogno di trovarsi nella stessa stanza per far avanzare un progetto.

Malorean Corp. sviluppava un ERP accessibile online: uno di quei programmi che finiscono per entrare praticamente in ogni operazione di un'azienda. Un cliente poteva essere una piccolissima impresa, con pochi dipendenti e necessità molto concrete. Un altro poteva essere un grande gruppo, con processi complessi, esigenze particolari e poca pazienza per le approssimazioni.

La clientela era soprattutto francese. C'erano anche alcuni utenti in Germania e Spagna, ma principalmente nelle filiali di gruppi già clienti in Francia. Hubert Félix Duca, alla guida dell'azienda, guardava più lontano. Avrebbe voluto vedere Malorean Corp. espandersi davvero in Europa.

Il problema è che un ERP non supera i confini con la stessa facilità di un sito vetrina. Le regole cambiano da un paese all'altro. Così come le procedure per le paghe, gli obblighi amministrativi e il modo di lavorare delle imprese. Hubert conosceva bene questi vincoli. Veniva dal settore della gestione, della contabilità e della finanza, non dallo sviluppo software. Per lui, una funzionalità aveva valore solo se i clienti potevano usarla senza ritrovarsi nei guai.

All'interno dell'azienda, i ruoli erano abbastanza chiari.

Due team di cinque sviluppatori ciascuno si occupavano di far evolvere il software. Alice e Bob, i due Product Manager, lavoravano con loro per comprendere le esigenze, scegliere le migliorie utili e dare una direzione al prodotto. Alice era stata sviluppatrice: poteva partecipare alle discussioni tecniche senza perdere il filo. Bob proveniva da una scuola di economia e commercio e affrontava i problemi da un'altra prospettiva. Avevano esperienze diverse, ma lavoravano molto bene insieme.

Un team commerciale cercava nuovi clienti e coltivava le relazioni esistenti, con il sostegno di una persona addetta al marketing. Cinque specialisti intervenivano nella prevendita e poi durante le implementazioni, di persona o a distanza. Non si limitavano a installare l'ERP: aiutavano le aziende a precisare le loro esigenze, ad adattare i processi e spesso continuavano a seguirle nei sei mesi successivi alla vendita.

L'assistenza clienti contava dodici persone. Sette seguivano i clienti più piccoli, cinque quelli grandi. Erano spesso le prime a sapere che una funzionalità obbligava un cliente a trovare un modo per aggirare il programma, anziché usarlo normalmente. Due giuristi si occupavano degli affari di Malorean Corp., ma contribuivano anche alla progettazione del prodotto, soprattutto quando una modifica riguardava norme delicate, come quelle sulle retribuzioni.

Tutto questo faceva funzionare l'azienda. Non alla perfezione, naturalmente. Un programma installato presso così tanti clienti accumula richieste, eccezioni, abitudini difficili da cambiare e funzioni che si mantengono perché manca il tempo di rifarle. Ma ciascuno sapeva più o meno che cosa doveva fare.

Poi ChatGPT aveva cominciato a comparire nelle conversazioni.

All'inizio non c'era un grande programma di trasformazione. Due persone fra le più curiose lo avevano provato qualche mese prima: Julie Morel,

delle vendite, e Lucas Martin, dello sviluppo. Avevano iniziato per le proprie necessità e poi avevano raccontato ciò che riuscivano a fare.

Per Julie, il vantaggio era evidente. Preparava un'email delicata, chiedeva una formulazione più chiara, riorganizzava gli appunti prima di un appuntamento, trasformava una serie di idee nella scaletta di una presentazione. A volte preparava anche il contenuto di alcune slide. Non era ChatGPT a decidere che cosa dire al cliente. Julie forniva le informazioni, rivedeva le proposte, correggeva le formulazioni e inviava solo ciò che era pronta a sostenere.

Non sosteneva che lo strumento conoscesse i clienti meglio di lei. Era semplicemente contenta di non dover passare mezz'ora a dare forma a qualcosa che sapeva già.

Philippe Garnier, il suo collega, osservava tutto con distacco. Commerciale esperto, abituato ai grandi clienti, conosceva le persone dietro i contratti, i disaccordi che non comparivano in nessun verbale, le piccole frasi capaci di cambiare il senso di una riunione.

— Dovresti provarlo per le tue email — gli aveva detto Julie una mattina. — Anche solo per sistemarle.

— Le email so già scriverle.

Non lo aveva detto con cattiveria. Semplicemente non capiva perché affidare a uno strumento una parte del proprio lavoro, quando il suo valore stava proprio, secondo lui, nella conoscenza dei clienti e nel modo di parlare con loro.

Julie non aveva insistito. Doveva preparare un appuntamento.

Lucas, invece, usava ChatGPT in modo completamente diverso. Sviluppatore esperto che viveva a Lione, non cercava soltanto di risparmiare qualche minuto nelle attività ripetitive. Passava molto tempo a sperimen-

tare, a porre domande sempre più impegnative, a ricominciare quando una risposta non lo convinceva. Julie cercava soprattutto un risultato utile; Lucas voleva scoprire fin dove si potesse arrivare.

Le impressioni dei primi utenti erano infine arrivate a Hubert. A una ventina di euro al mese per persona, il calcolo sembrava semplice: sarebbe bastato recuperare anche poco tempo sui compiti più noiosi per giustificare ampiamente l'investimento. Così Malorean Corp. aveva deciso di mettere ChatGPT a disposizione di tutti.

Non era una rivoluzione annunciata con presentazioni in grande stile. Ciascuno poteva usarlo quando gli sembrava utile. Alcuni lo aprivano ogni giorno, altri raramente e Philippe mai. Per la maggior parte delle persone, l'IA rimaneva un assistente: le spiegavi ciò che volevi, lei proponeva qualcosa, tu verificavi e poi decidevi.

La vita aziendale continuava.

Il mercoledì, durante una discussione tra i team di Prodotto e Sviluppo, era riemersa una vecchia questione: la gestione delle ferie e dei permessi.

Quel modulo non piaceva davvero a nessuno. Era vecchio, rigido e offriva poche opzioni. Funzionava abbastanza da continuare a essere usato, ma non così bene da smettere di ricevere critiche.

Un dipendente che volesse chiedere giorni di ferie doveva usare il computer di lavoro. Non poteva farlo facilmente dal telefono, una domenica sera, mentre pensava alle vacanze. E il calcolo delle assenze aveva un difetto ancora più imbarazzante: il modulo non teneva correttamente conto delle festività. Per una richiesta dal lunedì al venerdì, se il mercoledì era festivo, poteva scalare cinque giorni anziché quattro.

L'assistenza conosceva il problema. Anche i commerciali. Per alcuni clienti quel modulo era diventato motivo ricorrente di irritazione. Lo si manteneva perché esisteva, non perché rispondesse davvero alle aspettative.

— Dovremmo rifarlo come si deve — aveva riassunto qualcuno.

Nessuno era contrario. Ma «rifarlo come si deve» significava ripensare l'interfaccia, correggere le regole di calcolo, renderlo utilizzabile da telefono e sostituire una parte di ciò che era invecchiato.

Si parlava di un lavoro di circa due mesi. C'erano altre priorità. Ancora una volta, la questione era finita nella lista delle cose da fare.

Il venerdì pomeriggio, al termine di una videoconferenza, Lucas chiese qualche minuto.

— Ho qualcosa da mostrarvi.

Non aveva anticipato l'argomento. Nessuno sapeva che aveva lavorato al modulo delle ferie dopo la discussione del mercoledì.

Condivise lo schermo. Si aprì un'interfaccia chiara e moderna, con le informazioni essenziali esattamente dove ci si aspettava di trovarle.

Lucas mostrò come inviare una richiesta di ferie. Poi ridusse la finestra alla larghezza dello schermo di un telefono. L'interfaccia si adattò. Si poteva proseguire, scegliere le date e confermare la richiesta senza tornare al computer.

Passò a un secondo esempio: una settimana dal lunedì al venerdì, con una festività il mercoledì.

Il conteggio mostrava quattro giorni.

Modificò le date. Il risultato venne ricalcolato. Tornò alla vista per telefono, scorse alcune schermate e continuò la dimostrazione. Non era una sequenza di schermate statiche: i moduli reagivano, i calcoli venivano eseguiti, il percorso funzionava.

All'inizio, alcuni colleghi avevano posto le solite domande: dove si trovava un'opzione, che cosa succedeva cambiando una data? Poi le domande erano cessate.

Nelle piccole finestre della videoconferenza, le espressioni erano cambiate.

Mercoledì avevano parlato di due mesi di lavoro. Venerdì Lucas mostrava loro una prima versione funzionante. Non aveva chiesto aiuto. Non aveva avvisato nessuno. E per il momento non spiegava come ci fosse riuscito.

— Aspetta — disse finalmente qualcuno. — È il modulo di cui abbiamo parlato l'altro ieri?

Lucas confermò.

Cadde il silenzio.

Sugli schermi tutti guardavano la stessa cosa: un lavoro che pensavano di dover pianificare nell'arco di settimane, comparso in due giorni.

Nessuno capiva come fosse stato possibile.

2 – L'IA ha moltiplicato la capacità di sviluppo software

Per decenni, buona parte dell'organizzazione dello sviluppo software si è costruita intorno a un vincolo semplice: **produrre software costa molto**.

Trasformare un'idea in un prodotto richiedeva tempo, competenze rare e molta coordinazione. Occorreva definire il perimetro di una funzionalità, suddividerla, stabilirne la priorità, svilupparla, testarla, revisionarla e distribuirla. Il numero di sviluppatori disponibili, la loro esperienza e il tempo che potevano dedicarle limitavano inevitabilmente le iniziative di un'azienda.

Questa scarsità ha plasmato le organizzazioni. Abbiamo imparato a proteggere il tempo degli sviluppatori, a prepararne il lavoro, ridurre le interruzioni, organizzare i backlog e discutere a lungo le priorità prima di cominciare a costruire. Una buona parte del modo moderno di gestire i prodotti è nata in questo contesto: bisognava evitare di sprecare risorse di ingegneria costose per un'idea sbagliata.

L'arrivo degli assistenti di programmazione e poi degli agenti capaci di svolgere compiti più ampi comincia a cambiare questa equazione.

Non si tratta più soltanto di un completamento automatico migliore

I primi strumenti di IA per sviluppatori potevano essere visti come una nuova generazione di strumenti per la produttività: completare una funzione, spiegare un frammento di codice, scrivere un test, suggerire un refactoring.

La generazione successiva cambia radicalmente la natura del lavoro. Un agente può ricevere un compito, esplorare una base di codice, modificare più file, eseguire test, correggere alcuni errori e consegnare una modifica

pronta per la revisione. Uno sviluppatore può dunque delegare una parte importante dell'esecuzione e, soprattutto, avviare e far progredire molto più lavoro di prima.

La conseguenza non è solo che «lo sviluppatore programma più in fretta».

La quantità di lavoro software che una singola persona può avviare e portare avanti aumenta enormemente.

Alcune aziende iniziano a pubblicare risultati impressionanti. AMPECO, che descrive il proprio passaggio a un sistema di ingegneria AI-native, dichiara una velocità di sviluppo quattro volte superiore e il 50% di bug in meno per user story. Non dobbiamo trattare questi numeri come una media di mercato: sono risultati dichiarati da un'azienda, in un contesto particolare. Ma illustrano la portata del cambiamento ricercato. Non si tratta più soltanto di risparmiare qualche minuto scrivendo una funzione, bensì di aumentare notevolmente la capacità esecutiva di un team. [1]

Questo risultato spettacolare, rivendicato da un'azienda, non è l'unico tipo di indizio disponibile. Ho vissuto qualcosa di simile usando Claude Code. In quattro giorni e oltre 180 commit ho costruito una prima versione di un prodotto con schermate, API e test. Con i miei metodi precedenti, stimo che avrei impiegato uno o due mesi. Ho raccontato l'esperienza su Malorean.net nel luglio 2026. Naturalmente non consente di generalizzare un aumento di produttività: mostra che cosa può realizzare oggi un PM con un'esperienza precedente nello sviluppo, ma anche le difficoltà legate alla qualità, alla manutenibilità e al controllo del codice generato. [7]

Tutto questo cambia la scala della domanda.

Se l'IA facesse risparmiare soltanto qualche punto percentuale nella scrittura del codice, l'organizzazione potrebbe rimanere uguale e assorbire il vantaggio. Quando invece la capacità produttiva aumenta molto più rapidamente, gli altri vincoli diventano relativamente più importanti.

Accelerare la scrittura del codice non basta ad accelerare l'azienda

Un team software non è una catena il cui unico anello sia scrivere codice.

Prima di costruire occorre comprendere un problema, decidere se valga un investimento, scegliere una direzione e chiarire alcuni vincoli. Dopo bisogna revisionare, testare, mettere in sicurezza, distribuire, osservare e mantenere. Nel frattempo, bisogna continuare a prendere decisioni.

Se soltanto uno di questi passaggi accelera molto, non fa che trasferire più lavoro agli altri.

Il codice può essere prodotto più rapidamente senza che la comprensione dei clienti progredisca allo stesso ritmo. I prototipi possono moltiplicarsi senza che cresca la capacità di decidere quali meritino di essere sviluppati. Più modifiche possono arrivare alla revisione senza che aumenti nella stessa misura l'attenzione umana disponibile a esaminarle.

Il problema, quindi, non è che l'IA renda improvvisamente inutili le altre attività.

È che **sposta i colli di bottiglia**.

Ciò che prima era nascosto dalla lentezza dell'esecuzione diventa più visibile: la qualità del problema scelto, la chiarezza dell'intenzione, le decisioni, la validazione, la coerenza e la capacità di imparare dopo la consegna.

Produrre di più non significa creare più valore

Questa distinzione diventa fondamentale.

Quando produrre era difficile, la quantità di software consegnato poteva sembrare un buon indicatore della capacità di un'organizzazione di creare valore. Era già una scorciatoia imperfetta. Con l'IA diventa pericolosa.

Un team può ora generare più codice, più pull request, più prototipi e più funzionalità senza migliorare il prodotto. Può addirittura peggiorare

il rendimento complessivo se il volume aggiuntivo aumenta i difetti, il lavoro da rifare, i costi o la complessità della manutenzione.

In altre parole:

saper costruire non equivale a saper scegliere ciò che vale la pena costruire.

Ecco perché le metriche di attività sono ancora meno sufficienti. Il numero di modifiche, la quantità di codice o il semplice utilizzo di uno strumento di IA non dicono se un team abbia creato più valore.

Canva offre un esempio interessante di questo spostamento verso l'impatto: ha sviluppato un sistema che consente ai team di Prodotto, Ingegneria e Dati di stimare l'impatto commerciale degli esperimenti e collegare più direttamente le decisioni locali alle conseguenze per il business. [2]

La domanda cambia allora natura.

Prima:

quanto possiamo costruire?

Adesso:

tra tutto ciò che siamo ormai in grado di costruire, che cosa vale davvero la pena realizzare?

> ***Il problema della produttività del software ha più di quarant'anni***
>
> *Nel 1986 Fred Brooks presentò* No Silver Bullet*, pubblicato su* IEEE Computer *nel 1987. Distingueva le difficoltà* ***accidentali*** *dello sviluppo software — linguaggi, strumenti, traduzione di un progetto in codice — dalla sua complessità* ***essenziale****: capire, progettare, articolare e far evolvere un sistema. Secondo Brooks, nessun singolo progresso tecnico o*

> *manageriale poteva promettere di moltiplicare per dieci in un decennio la produttività, l'affidabilità e la semplicità dello sviluppo. [3]*
>
> *Il saggio fu inserito nel 1995 nell'edizione commemorativa di* The Mythical Man-Month*, uscito originariamente nel 1975. [4] L'IA generativa può potenzialmente cambiare la scala dell'esecuzione, ma non garantisce che il problema scelto sia quello giusto. È una lettura attuale delle idee di Brooks,* ***non una sua previsione sugli agenti di IA****.*

Il nuovo collo di bottiglia è il giudizio

Man mano che eseguire diventa meno costoso, altre risorse acquistano relativamente più valore: conoscere il cliente, comprendere il problema, disporre del contesto strategico, saper esprimere un'intenzione chiara, valutare i rischi, avere sensibilità di prodotto, saper dire di no e assumersi la responsabilità delle decisioni.

In un articolo pubblicato su Malorean.net nel marzo 2026 sostenevo questa idea: man mano che gli strumenti tecnici diventano più accessibili, il vantaggio competitivo si sposta verso la comprensione del contesto — professioni, utenti, clienti, vincoli tecnici, concorrenza e normative. Si tratta di un'analisi strategica, non di un risultato sperimentale. Chiarisce che cosa significa esercitare il proprio giudizio: comprendere abbastanza bene una situazione da scegliere la cosa giusta da costruire, e non soltanto saperla produrre. [8]

L'IA rende più accessibile il «come».

Ma non rende automaticamente evidente il «perché».

Permette di realizzare una soluzione prima che l'organizzazione abbia finito di comprendere il problema. Permette di soddisfare rapidamente una richiesta di un cliente che l'azienda avrebbe dovuto rifiutare perché

estranea al proprio mercato. Permette di moltiplicare le varianti di una funzionalità senza sapere quale porti davvero a un risultato migliore.

Più costruire diventa facile, più diventa importante la disciplina di scegliere ciò che non costruiremo.

Ramp illustra bene il principio: il suo CPO insiste sulla necessità di definire gli ICP prioritari, i problemi che l'azienda vuole risolvere particolarmente bene e di accettare di rifiutare certe richieste — anche redditizie — quando spingono il prodotto in un'altra direzione. [5]

Nel luglio 2026 Marty Cagan ha definito questo scarto il *paradosso della produttività dell'IA*. Secondo la sua analisi, costruire più velocemente non risolve il problema delle idee che non avrebbero mai meritato di essere sviluppate. Distingue quindi ciò che costruiamo per imparare da ciò che consegniamo per ottenere un risultato. È l'analisi di un professionista del prodotto, non una misurazione che dimostri un effetto uguale in tutte le aziende. [6]

Il paradosso dell'IA è dunque questo:

aumenta enormemente la nostra capacità di agire e rende quindi ancora più importante la scelta delle nostre azioni.

Una trasformazione del sistema, non solo degli strumenti

Si potrebbe rispondere a questo cambiamento dando semplicemente un assistente di programmazione a ogni sviluppatore e chiedendo ai team di lavorare come prima.

Questo può produrre miglioramenti locali.

Ma con l'aumentare di tali vantaggi, l'organizzazione incontra gli altri propri limiti: la capacità di comprendere, decidere, coordinare, verificare e assorbire tutto ciò che viene prodotto.

A quel punto non si tratta più soltanto di adottare uno strumento.

La questione diventa organizzativa.

Un'organizzazione progettata quando l'ingegneria software era la risorsa più scarsa non è necessariamente ancora ottimale quando questa scarsità diminuisce.

Diventa inevitabile una domanda:

che cosa succede a un'organizzazione progettata per proteggere una risorsa scarsa — la capacità di sviluppo software — quando questa diventa molto più abbondante?

Bibliografia

[1] How we built an AI-native engineering system – AMPECO

- URL: https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Testimonianza su un'organizzazione di ingegneria ricostruita intorno agli agenti, con un forte aumento della capacità dichiarato dall'azienda.

[2] Measuring Commercial Impact at Scale – Canva

- URL: https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/
- Spiega come Canva colleghi gli esperimenti dei team di Prodotto e Ingegneria a una stima accessibile del loro impatto commerciale.

[3] No Silver Bullet: Essence and Accidents of Software Engineering – Fred Brooks

- URL: https://www.cgl.ucsf.edu/Outreach/pc204/NoSilverBullet
- Saggio presentato nel 1986 e pubblicato nel 1987, che distingue la complessità essenziale dalle difficoltà accidentali dello sviluppo software.

[4] The Mythical Man-Month, edizione commemorativa, capitolo 16 – Fred Brooks

- URL: https://www.oreilly.com/library/view/mythical-man-month-the/0201835959/ch16.xhtml
- Sommario editoriale dell'edizione del 1995 che conferma l'inserimento successivo di No Silver Bullet; è disponibile soltanto un'anteprima del capitolo.

[5] 5 insights from Ramp's CPO on building for hypergrowth – Ramp

- URL: https://ramp.com/blog/5-insights-from-ramps-cpo-reddit-ama
- Il CPO di Ramp sottolinea l'importanza di concentrarsi sugli ICP prioritari e di rifiutare le richieste che disperdono le energie.

[6] The AI Productivity Paradox – Marty Cagan, Silicon Valley Product Group (2026)

- URL: https://www.svpg.com/the-ai-productivity-paradox/
- Analisi del luglio 2026 che distingue la maggiore velocità consentita dall'IA dai risultati creati; riflessione professionale, non dimostrazione causale.

[7] Quatre jours de vibe coding dans la peau d'un PM rouillé – Laurent Deséchalliers, Malorean.net (2026)

- URL: https://malorean.net/articles/2026-07-28-quatre-jours-de-vibe-coding-dans-la-peau-d-un-pm-rouille.it.html
- Esperienza personale con Claude Code: velocità di creazione del prodotto, difficoltà tecniche, test e controllo della qualità; né studio controllato né prova generalizzabile.

[8] Dans le logiciel, l'avantage ne sera plus la technologie. Ce sera la compréhension du contexte – Laurent Deséchalliers, Malorean.net (2026)

- URL: https://malorean.net/articles/2026-03-23-dans-le-logiciel-lavantage-ne-sera-plus-la-technologie-ce-sera-la-comprehension-du-contexte.it.html
- Saggio strategico dell'autore sullo spostamento della scarsità verso la comprensione del contesto del cliente, professionale, normativo e tecnico; non una dimostrazione quantitativa.

3 – Malorean Corp. – Il giorno dopo

— Ho usato l'IA.

Lucas lo aveva detto quasi con naturalezza, come se bastasse a spiegare ciò che tutti avevano appena visto.

Stava ancora condividendo lo schermo. Il nuovo modulo per ferie e permessi occupava gran parte della finestra. Qualche minuto prima aveva mostrato come chiedere un'assenza dal telefono e come il software tenesse conto dei giorni festivi nel calcolo.

— Aspetta — intervenne Alice. — Quando dici che hai usato l'IA, intendi per aiutarti a scrivere il codice?

— Sì. Beh, non solo. Per progettare il modulo, capire il codice esistente, sviluppare le schermate, fare i calcoli, scrivere i test… Un po' per tutto, in realtà.

— E hai fatto tutto questo da mercoledì?

— Da mercoledì pomeriggio. Ho finito le ultime cose stamattina.

Alice osservò i volti sullo schermo. Alcuni sorridevano. Altri sembravano cercare di capire che cosa fosse sfuggito loro.

— Ma non avevi delle specifiche — osservò.

— No. Ne avevamo parlato mercoledì. Conoscevo il modulo e i suoi problemi. Per prima cosa ho chiesto all'IA di analizzare quello che c'era. Poi siamo andati avanti un pezzo alla volta.

— E funziona tutto?

Lucas ridacchiò.

— Tutto non lo so. Quello che vi ho mostrato sì. Ci saranno sicuramente altre cose da verificare.

Thomas, che lavorava nello stesso team, intervenne per la prima volta.

— E com'è il codice, sotto?

— Puoi guardarlo. Ho fatto parecchi test.

— Lo guarderò — rispose Thomas.

Non sembrava particolarmente colpito. O forse lo era, ma non voleva mostrarlo.

Hubert Félix Duca era rimasto in silenzio fino a quel momento.

Aveva assistito a abbastanza dimostrazioni da sapere che un programma capace di funzionare per dieci minuti non è necessariamente pronto per essere usato da centinaia di aziende. Conosceva le eccezioni, le esigenze particolari dei grandi clienti, i problemi che emergono soltanto quando il software arriva nelle loro mani.

Ma ricordava anche ciò che aveva sentito mercoledì.

Due mesi.

E ora aveva davanti qualcosa che somigliava già molto a ciò che avevano immaginato.

— Lucas, mi ripeti quanto tempo hai dedicato a questo lavoro? — chiese.

— Due giorni, più o meno.

Hubert si sporse leggermente verso lo schermo.

— D'accordo.

Non trovò altro da aggiungere.

La riunione terminò qualche minuto dopo. Tutti tornarono alle proprie attività, ma nessuno riprese davvero a parlare degli impegni di quel pomeriggio.

Il sabato mattina, mentre preparava il caffè, Alice ripensò alla dimostrazione.

Ciò che la turbava non era soltanto la velocità di Lucas. Era il fatto che non avesse avuto bisogno di nessuno.

Conosceva bene il proprio lavoro di Product Manager. Prima di avviare uno sviluppo cercava di capire il problema, confrontava le richieste, parlava con i clienti e lavorava insieme agli sviluppatori per trovare una soluzione ragionevole.

Su un argomento come il modulo delle ferie, probabilmente avrebbe organizzato diversi incontri. Avrebbe contribuito a chiarire le regole, scegliere come gestire i casi particolari e decidere che cosa sviluppare per primo.

Lucas non aveva fatto niente di tutto questo.

Aveva ascoltato una discussione, preso l'iniziativa e costruito una soluzione.

In parte lo ammirava. Ma si chiedeva anche che cosa sarebbe successo se tutti gli sviluppatori avessero cominciato a lavorare così.

Eppure, negli ultimi mesi, la sua stessa vita lavorativa era cambiata parecchio.

All'inizio Alice usava l'IA per riassumere verbali o riformulare documenti. Poi aveva cominciato ad affidarle compiti più importanti. Aveva automatizzato una parte dell'elaborazione dei commenti dei clienti, la preparazione di alcuni riepiloghi, il raggruppamento di richieste simili e diverse attività amministrative che prima le portavano via ore.

Ora poteva concludere in pochi minuti un lavoro che in passato occupava buona parte della mattinata.

Non rimpiangeva nessuno di quei compiti.

Anzi.

Finalmente aveva più tempo per parlare con i clienti, capire i problemi nascosti dietro le loro richieste e riflettere su ciò che valesse davvero la pena costruire.

L'IA non aveva reso meno interessante la sua professione.

L'aveva liberata da una parte di ciò che le impediva di svolgerla bene.

Perché, allora, gli sviluppatori avrebbero dovuto continuare a lavorare esattamente come prima?

Alice posò la tazza.

La domanda le sembrava ovvia.

Molto meno la risposta.

Anche Hubert aveva passato buona parte del fine settimana a riflettere.

La sua azienda aveva sempre avuto più idee che capacità di realizzarle. Alcune funzionalità aspettavano da mesi. Certi miglioramenti chiesti dai clienti venivano continuamente rimandati. E c'era il progetto di espansione europea, che richiedeva ancora molto lavoro sul software.

L'ostacolo lo conosceva a memoria: non avevano abbastanza tempo per fare tutto.

Venerdì Lucas aveva forse dimostrato che quell'ostacolo non era più così insormontabile.

Se un progetto stimato in due mesi poteva produrre una prima versione in due giorni, quanti altri progetti diventavano improvvisamente possibili?

In Malorean Corp. lavoravano dieci sviluppatori. Hubert aveva provato a immaginare che cosa sarebbe successo se tutti avessero potuto procedere così velocemente.

Poi aveva immaginato dieci sviluppatori che avviavano ognuno le proprie funzionalità, senza coordinarsi.

La seconda immagine gli piaceva decisamente meno.

Aveva pensato di chiamare Alice già il sabato. Si era trattenuto.

Domenica sera le aveva inviato soltanto un messaggio.

Hai un momento domani mattina? Vorrei tornare su quello che è successo venerdì.

La risposta arrivò pochi minuti dopo.

Sì. Anch'io.

Lunedì mattina, alle nove, Alice e Hubert si incontrarono in videoconferenza.

— Allora? — chiese Hubert.

— Allora cosa?

— Hai capito cos'è successo venerdì?

Alice sorrise.

— Dal punto di vista tecnico, più o meno. Lucas ha usato l'IA per sviluppare molto più rapidamente di quanto avremmo immaginato.

— E il resto?

— È proprio questo il problema.

Hubert annuì.

— Ci ho pensato tutto il fine settimana. Ci manca capacità da anni. Rifiutiamo richieste, rimandiamo progetti, passiamo il tempo a scegliere che cosa non fare. E adesso Lucas ci tira fuori in due giorni qualcosa che avevamo valutato due mesi.

— Una prima versione — precisò Alice. — Non sappiamo ancora se sia pronta per andare in produzione.

— Sì, lo so. Ma anche se ci volessero altre due settimane per finirla, cambierebbe moltissime cose.

Alice non poteva contraddirlo.

— Quello che mi preoccupa — continuò Hubert — è che abbia fatto tutto senza passare da te.

Alice non rispose subito.

— Ha dato fastidio anche a me — ammise.

— Pensi che dovremmo ricordargli come lavoriamo?

— Ci ho pensato.

— E?

— E credo che sarebbe un errore.

Hubert aggrottò la fronte.

— Ma se comincia a decidere da solo che cosa sviluppare, rischiamo qualche brutta sorpresa.

— Certo. Però guarda che cos'è successo. Mercoledì diciamo che bisogna rifare il modulo delle ferie. Se avessimo seguito il nostro modo abituale di lavorare, avrei cominciato a raccogliere le esigenze, avremmo organizzato riunioni, preparato il progetto, diviso il lavoro… Probabilmente sarebbero passate settimane prima di mostrare qualcosa.

— Non è necessariamente tempo sprecato.

— No. E non sto dicendo che dobbiamo smettere di capire le esigenze. Ma Lucas ci ha mostrato che poteva andare avanti senza aspettare tutto questo.

— Quindi proponi di lasciare che siano gli sviluppatori a decidere da soli?

— No, proprio il contrario.

Alice rifletté per qualche secondo.

— Credo che stiamo confondendo due cose: decidere che cosa dobbiamo fare e decidere come farlo.

— Sono collegate.

— Sì. Ma questo non significa che debba intervenire in ogni decisione.

Si raddrizzò leggermente sulla sedia.

— Sai, il mio lavoro è cambiato moltissimo negli ultimi mesi. Uso l'IA tutti i giorni. Ho automatizzato tante cose che prima facevo a mano. Passo molto meno tempo a preparare documenti o a organizzare informazioni.

— Lo so, me ne hai parlato.

— Ma forse non ti rendi conto di quanto sia cambiato davvero. Oggi ho molto più tempo per ciò che conta: capire i clienti, individuare i problemi e decidere dove investire le nostre energie.

Hubert ascoltava.

— Prima una parte del mio lavoro consisteva nel far avanzare i progetti nell'organizzazione. Preparare i ticket, chiarire le richieste, rispondere alle domande, coordinare le conversazioni. E se domani gli sviluppatori avessero molto meno bisogno di me per queste cose?

— Avresti più tempo.

— Esatto. E anche loro.

Fece una pausa.

— Accettiamo che l'IA trasformi la mia professione. Perché dovremmo volere che quella degli sviluppatori rimanga identica?

Hubert si appoggiò allo schienale.

— D'accordo. Però qualcuno deve comunque decidere che cosa costruiamo.

— Ovviamente. E soprattutto perché lo costruiamo.

— Allora come immagini le cose?

Alice esitò.

Non aveva preparato alcuna presentazione. Non aveva un modello organizzativo da proporre, né un nuovo processo da introdurre.

Aveva soltanto un'intuizione.

— Mi chiedo se non dovremmo provare qualcosa.

Hubert attese che proseguisse.

— Potremmo affidare un progetto a Lucas. Un problema reale di un cliente. E lasciargli molta più libertà di risolverlo.

— Senza specifiche?

— Senza dargli una soluzione già pronta. Gli spieghiamo il problema, lo mettiamo in contatto con chi lo conosce e lo lasciamo cercare.

— E tu?

— Rimango disponibile. Lo aiuto a capire le esigenze, metto alla prova le sue idee ed evito che sviluppi qualcosa solo perché è in grado di farlo. Ma non gli indico ogni singolo passo.

Hubert rifletté.

— Vuoi quindi che faccia una parte del lavoro di Product Manager?

— Forse. Oppure potremmo semplicemente smettere di dividere sistematicamente il lavoro fra noi.

— E se non funziona?

— Lo scopriremo.

Hubert sorrise.

— Come metodo è un po' debole.

— Lo so.

Anche Alice sorrise.

— Però preferisco provarlo su un progetto limitato anziché passare tre mesi a discutere di una nuova organizzazione.

Hubert non rispose subito.

Poi sembrò ricordarsi di qualcosa.

— In effetti abbiamo un problema segnalato da settimane da alcuni clienti del settore alimentare.

Alice annuì. Sapeva benissimo di quale si trattasse.

L'ERP di Malorean Corp. gestiva il magazzino e le operazioni di produzione, ma non permetteva di tracciare correttamente i lotti di materie prime usati per realizzare i prodotti finiti.

In un'azienda alimentare, quando un fornitore segnalava un lotto difettoso di materia prima, era necessario identificare rapidamente i prodotti fabbricati con quel lotto.

In quel momento i clienti di Malorean Corp. dovevano ricostruire le informazioni da soli.

— Ho ricevuto un altro riscontro proprio la settimana scorsa — spiegò Hubert. — Un cliente mi ha detto che lui e il responsabile della qualità hanno passato un'intera giornata a ritrovare i prodotti interessati da un'allerta di un fornitore.

— Sì. L'assistenza ci ha segnalato diversi casi simili.

— Pensi che Lucas potrebbe occuparsene?

Alice rifletté.

Il progetto era più delicato del modulo delle ferie. Non bastava creare alcune schermate. Bisognava capire che cosa succedesse davvero negli stabilimenti, come venissero usate le materie prime e come i clienti identificassero i loro prodotti.

Ma era proprio questo che la interessava.

— Credo che potremmo affidargli il problema, sì.

— Pensi che conosca abbastanza il settore?

— Probabilmente no.

Hubert alzò le sopracciglia.

— Rassicurante.

— Nemmeno io conosco tutti i dettagli. Però so con chi parlare. Abbiamo clienti che conoscono benissimo i loro vincoli. E persone nell'assistenza e nell'implementazione che vedono i problemi ogni giorno. Lucas può andare da loro.

— Vuoi che parli direttamente con i clienti?

— Perché no?

— Di solito è il tuo lavoro.

— Appunto.

Hubert non sembrava del tutto convinto.

— E se costruisce qualcosa che non risponde alle esigenze?

— Il mio compito è aiutarlo a evitarlo. Ma non scrivendogli cinquanta pagine di specifiche.

— E se volesse mettere in produzione la sua soluzione?

— Dovrà essere verificata e approvata, come qualsiasi altra modifica al software. Più libertà non significa autorizzarlo a correre tutti i rischi.

Hubert rimase in silenzio per qualche istante.

— Sai che cosa mi preoccupa? — chiese infine. — Non soltanto che possa sbagliare. Mi preoccupa che non si capisca più chi fa che cosa.

— Preoccupa anche me.

— E vuoi provare lo stesso?

— Sì.

Si prese un momento prima di aggiungere:

— Perché l'alternativa è continuare esattamente come prima, chiedendo semplicemente agli sviluppatori di programmare più in fretta. E non sono sicura che sia questo ciò che Lucas ci ha mostrato venerdì.

Continuarono a parlare per una ventina di minuti.

Hubert voleva evitare di disturbare i progetti in corso. Alice non voleva trasformare l'esperimento in una grande iniziativa annunciata a tutta l'azienda ancora prima di cominciare.

Decisero quindi di mantenere le cose semplici.

Lucas avrebbe lavorato al problema della tracciabilità dei lotti. Avrebbe avuto la libertà di esplorare soluzioni e incontrare direttamente le persone in grado di aiutarlo a capire le esigenze.

Alice sarebbe stata il suo riferimento per il Prodotto. Non gli avrebbe preparato un backlog dettagliato, ma sarebbe rimasta coinvolta nelle decisioni importanti.

Avrebbero cominciato con un perimetro limitato, senza promettere a nessun cliente una data di consegna.

E soprattutto si sarebbero presi il tempo di osservare ciò che sarebbe accaduto.

— Dovremmo comunque stabilire come capiremo se l'esperimento è riuscito — osservò Hubert.

— Sì. E non voglio che misuriamo soltanto quante funzionalità avrà sviluppato.

— Allora che cosa vuoi misurare?

— Prima di tutto, se riesce a capire il problema e a proporre qualcosa di utile. Poi, fino a che punto può avanzare senza che dobbiamo organizzargli tutto il lavoro.

— E quanto tempo richiede a noi.

— Anche quello.

Hubert guardò l'orologio.

— Bene. Proviamo.

Alice sorrise, senza sentirsi del tutto rassicurata.

— Proviamo.

— Gliene parli oggi?

— Sì.

La videoconferenza terminò.

Alice rimase per qualche istante davanti allo schermo. Aprì la messaggistica e cercò Lucas.

Cominciò a scrivere:

Ciao Lucas. Hai un momento oggi? Vorrei proporti un progetto un po' diverso da quelli di cui ci occupiamo di solito.

Rilesse il messaggio prima di inviarlo.

Per la prima volta da molto tempo avrebbe affidato un problema a uno sviluppatore senza avere già cominciato a immaginarne nei dettagli la soluzione.

Non sapeva ancora se fosse una buona idea.

E non era più così sicura che il suo ruolo consistesse nel saperlo prima di lui.

4 – Stanno emergendo nuovi modelli organizzativi

L'IA permette di eseguire molto più lavoro. Ma come stanno riorganizzando le aziende i team e le decisioni? Le risposte sono diverse.

Alcune rafforzano il Product Management intorno agli sviluppatori. Altre ampliano il ruolo di questi ultimi, danno più strumenti alle figure non tecniche oppure riorganizzano il lavoro intorno a piccoli team assistiti dagli agenti.

La divisione tradizionale del lavoro non è più scontata.

Eppure il modello tradizionale aveva una logica: il Product Management concentrava gran parte del contesto relativo ai clienti e al business, il Design dava forma all'esperienza e l'Engineering trasformava queste intenzioni in software. Ticket, specifiche, passaggi di consegne e riunioni consentivano di sincronizzare le professioni intorno a una capacità di sviluppo limitata.

Questa organizzazione non diventa improvvisamente sbagliata.

Ma quando aumenta la capacità di esecuzione, alcune aziende cominciano a ridistribuire diversamente persone, contesto e lavoro.

Non si tratta ancora di cercare «il modello giusto».

Si tratta di osservare le principali risposte che stanno emergendo.

Rafforzare il Prodotto intorno agli sviluppatori

La risposta più intuitiva consiste nel mantenere una separazione relativamente classica tra Prodotto e Ingegneria, aumentando però la capacità di Product Management a disposizione degli sviluppatori.

Gamma offre un esempio interessante. L'azienda dichiara un rapporto di circa un Product Manager ogni quattro ingegneri, mentre il rapporto storico spesso citato è più vicino a uno ogni dieci. [1]

In questo modello, quindi, l'aumento della capacità di Ingegneria è accompagnato da una maggiore presenza del Product Management.

L'idea di base è semplice: se possiamo costruire più cose, dedichiamo più capacità anche a capire, stabilire le priorità e orientare quel lavoro.

Affidare allo sviluppatore una parte più ampia del progetto

Un'altra strategia consiste nell'ampliare il perimetro d'azione dell'ingegnere.

Lo sviluppatore non aspetta più necessariamente che qualcuno abbia tradotto completamente un problema in una soluzione e poi in ticket. Può partecipare di più alla comprensione delle esigenze, alla creazione dei prototipi, alle decisioni locali e alla verifica dei risultati.

Nella sua offerta di lavoro per Product Engineer, Linear descrive le responsabilità richieste: seguire una funzionalità dall'idea al prodotto consegnato e partecipare alla definizione del problema senza dipendere continuamente dal Product Management. [2]

Questo modello cerca di ridurre alcuni passaggi tra professioni, concentrando più contesto e capacità d'azione sulla stessa persona.

Non significa che ogni sviluppatore diventi un PM.

Descrive piuttosto un ampliamento del ruolo dell'Ingegneria su alcuni progetti.

Lasciare che le professioni superino i loro vecchi confini

Questo movimento non riguarda soltanto gli sviluppatori.

In uno studio sull'evoluzione dei ruoli, Figma rileva che il 64% dei partecipanti si identifica con almeno due ruoli e più di un terzo dichiara responsabilità che attraversano più professioni. Non è possibile attribuire questo dato soltanto all'IA, ma descrive confini professionali meno rigidi. [3]

Atlassian documenta inoltre flussi di lavoro in cui i designer modificano direttamente il codice reale del prodotto e aprono una pull request prima della revisione tecnica. [4]

Dal canto suo, Cursor descrive un'organizzazione in cui Prodotto, Design e Ingegneria partecipano più direttamente alla costruzione. [5]

La conclusione non è tanto «tutti fanno tutto», quanto che **sempre più persone intervengono al di fuori del loro perimetro tradizionale**.

Questa maggiore permeabilità solleva un'altra domanda: se diverse professioni possono agire sullo stesso terreno, come manteniamo chiare le responsabilità?

Ci torneremo.

Affidare più lavoro a piccoli team responsabili

Un'altra famiglia di risposte consiste nell'affidare una porzione di lavoro più ampia a meno persone.

AMPECO descrive brevi riunioni di allineamento, gli *alignment syncs*, nelle quali un ingegnere e un Product Owner definiscono l'intenzione prima che gli agenti si occupino di buona parte dell'analisi, dello sviluppo e dei test. [6]

Taito descrive un'organizzazione in cui il team approva un piano e poi una o due persone guidano la realizzazione con l'aiuto degli agenti. [7]

In questi modelli, un piccolo gruppo umano può avviare molto più lavoro rispetto al passato.

Il cambiamento osservabile è quindi una possibile riduzione dei passaggi tra professioni e la concentrazione della responsabilità di un progetto in una persona o in un piccolo gruppo.

Resta da stabilire quali decisioni possano prendere da soli e quando debbano rivolgersi al responsabile di un ambito.

Automatizzare parte delle verifiche

Una produzione più rapida mette sotto pressione anche i meccanismi di validazione.

Alcune aziende rispondono automatizzando più controlli o trattando le modifiche in modo diverso in funzione del rischio.

Meta utilizza un Diff Risk Score per stimare il rischio che una modifica provochi un incidente. [8]

Cloudflare definisce standard di Ingegneria che possono essere verificati automaticamente prima del merge. [9]

Taito combina inoltre regole deterministiche, agenti di revisione e criteri di rischio. [7]

Per ora, la conclusione importante è semplicemente che **la validazione non deve più essere necessariamente organizzata nello stesso modo per ogni modifica**.

Vedremo più avanti come combinare automazione, responsabilità umana e soglie oltre le quali occorre coinvolgere altri responsabili.

Fornire contesto anziché prescrivere ogni passaggio

Altre aziende vanno oltre: mettono in discussione il ticket dettagliato come principale strumento di coordinamento.

Linear critica esplicitamente il modello in cui il PM definisce il perimetro di un problema, lo registra in un sistema e aspetta che l'Ingegneria lo

realizzi. L'azienda insiste sulla necessità di rendere accessibili a persone e agenti i commenti dei clienti, le idee, la direzione strategica, le decisioni e il codice. [10]

Anche Cursor sottolinea la necessità di fissare obiettivi sufficientemente chiari da consentire più iniziativa locale. [5]

Il cambiamento osservato non è quindi la scomparsa di ogni forma di coordinamento.

È lo spostamento di una parte di esso dalle istruzioni dettagliate verso **un contesto più ricco e direttamente accessibile**.

Dedicheremo un capitolo a ciò che questo contesto deve contenere, a come mantenerlo e a come definire i limiti dell'autonomia.

Permettere a chi conosce la necessità di costruire

Alcune aziende consentono addirittura ai team che conoscono il problema di creare direttamente piccoli strumenti, senza passare ogni volta dalla coda di lavoro degli sviluppatori.

Shopify ha creato Quick, una piattaforma interna che permette a un numero maggiore di dipendenti di pubblicare rapidamente siti e piccoli strumenti. [11]

Vercel descrive un modello in cui persone non tecniche realizzano strumenti interni in un ambiente sicuro senza distogliere gli sviluppatori dalle priorità del prodotto principale. [12]

Anche Atlassian mostra designer che intervengono sul codice prima della revisione tecnica. [4]

In questi casi, l'IA non rende soltanto più rapidi gli sviluppatori.

A volte permette di affidare l'esecuzione **alla persona che conosce già il contesto della necessità**.

Un principio più antico dell'IA

Già nel 2018 Marty Cagan sosteneva quello che chiamava *empowered product teams*: team multidisciplinari ai quali la direzione affida problemi da risolvere, invece di un elenco di funzionalità da consegnare. Chiede loro di trovare soluzioni mettendo insieme le competenze di Prodotto, Design e Ingegneria e di rispondere del risultato, non della quantità prodotta. [13]

Insieme a Chris Jones ha sviluppato questo approccio nel libro *EMPOWERED* del 2020. Il modello non annunciava né professioni intercambiabili né la fine dei manager: richiedeva, al contrario, una visione condivisa, competenze solide e responsabili capaci di accompagnare i team. [14]

L'idea, dunque, precede gli agenti. Ciò che l'IA può cambiare è la capacità di realizzare più velocemente e direttamente alcune soluzioni che un team autonomo ha scelto di esplorare. **L'autonomia di Prodotto è un'ambizione precedente; l'IA ne cambia gli strumenti e i vincoli.**

Scelte che non convergono ancora

Queste risposte non sono fasi successive di un'unica trasformazione. Rafforzare il Prodotto e affidare più responsabilità di prodotto agli sviluppatori possono persino sembrare strategie contraddittorie. Eppure possono funzionare in aziende diverse, o coesistere nella stessa organizzazione.

Il loro punto comune è un altro: **quando eseguire diventa più accessibile, il modo di distribuire contesto, decisioni e responsabilità torna a essere una scelta**.

Resta da vedere che cosa provoca questo cambiamento nella vita quotidiana delle persone, al di là degli annunci e degli organigrammi.

Torneremo presto a Malorean Corp. per osservare gli effetti di questi modelli in una piccola azienda francese. Esamineremo poi le responsabilità e le regole necessarie per questa autonomia.

Bibliografia

[1] The bottleneck keeps shifting: What AI is changing about how we build – Atlassian / Gamma

- URL: https://www.atlassian.com/blog/how-we-build/the-bottleneck-keeps-shifting-what-ai-is-changing-about-how-we-build
- Descrive la maggiore presenza del Product Management in Gamma quando aumenta la capacità produttiva dell'Ingegneria.

[2] Product Engineer – Linear Careers

- URL: https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b
- Offerta di lavoro che presenta le responsabilità di un Product Engineer, dalla definizione del problema alla consegna.

[3] Are Roles and Responsibilities a Thing of the Past? – Figma

- URL: https://www.figma.com/blog/2025-shifting-roles-report/
- Documenta una maggiore permeabilità dei ruoli e l'ampliamento delle responsabilità oltre i confini professionali tradizionali.

[4] Designers' workflow for shipping code – Atlassian

- URL: https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code
- Descrive designer che intervengono direttamente sul codice del prodotto prima della validazione tecnica.

[5] How Cursor Is Navigating the New Shape of Product Craft – Atlassian / Cursor

- URL: https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Mostra un'organizzazione in cui Prodotto, Design e Ingegneria partecipano più direttamente alla realizzazione.

[6] How we built an AI-native engineering system – AMPECO

- URL: https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Descrive una struttura in cui poche persone definiscono l'intenzione prima di delegare agli agenti gran parte dell'esecuzione.

[7] Inside an AI-native engineering organization – Swarmia / Taito.ai

- URL: https://www.swarmia.com/blog/inside-an-ai-native-engineering-organization/
- Presenta un piccolo team che guida gli agenti, con gran parte dell'esecuzione concentrata in una o due persone.

[8] Diff Risk Score: AI-driven risk-aware software development – Meta

- URL: https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware-software-development-meta/
- Presenta un punteggio di rischio impiegato per differenziare le procedure di validazione secondo la criticità delle modifiche.

[9] How Cloudflare enforces engineering standards using AI – Cloudflare

- URL: https://blog.cloudflare.com/engineering-standards-enforcement/
- Mostra come standard di Ingegneria espliciti diventino controlli automatici nel flusso di sviluppo.

[10] Issue tracking is dead – Linear

- URL: https://linear.app/next
- Sostiene un modello in cui feedback, strategia, decisioni e codice condividono più contesto rispetto a un ticket isolato.

[11] Quick: An internal hosting platform for the AI era – Shopify

- URL: https://shopify.engineering/quick

- Presenta una piattaforma interna che consente a più dipendenti di costruire e pubblicare direttamente.

[12] Stopping the slow death of internal tools – Vercel

- URL: https://vercel.com/blog/stopping-the-slow-death-of-internal-tools
- Spiega come gli utenti aziendali possano costruire strumenti interni senza distogliere l'Ingegneria dalla roadmap principale.

[13] Empowered Product Teams – Marty Cagan, Silicon Valley Product Group (2018)

- URL: https://www.svpg.com/empowered-product-teams/
- Testo che distingue i team responsabili di risolvere problemi da quelli limitati alla consegna di funzionalità.

[14] EMPOWERED: Ordinary People, Extraordinary Products – Marty Cagan e Chris Jones (2020)

- URL: https://www.svpg.com/books/empowered-ordinary-people-extraordinary-products/
- Presentazione del libro sui team autonomi di Prodotto, gli obiettivi, il contesto strategico e il ruolo dei responsabili nella loro crescita.

5 – Malorean Corp. – Ciò che possiamo fare e ciò che dobbiamo fare

Il martedì mattina Lucas esordì con una domanda che sorprese Alice.

— Posso parlare con il cliente?

Il giorno prima aveva ricevuto le poche relazioni conservate dall'assistenza sul problema della tracciabilità dei lotti. I messaggi parlavano di numeri di produzione, materie prime, fogli Excel e segnalazioni dei fornitori. Aveva passato la serata a organizzarli con l'aiuto dell'IA.

— Certamente — rispose Alice. — Perché?

— Perché non capisco esattamente che cosa gli manca.

Alice sorrise. Una settimana prima si sarebbe probabilmente occupata lei dell'incontro, per poi riferire le conclusioni agli sviluppatori. Adesso aveva promesso di lasciare a Lucas la gestione del progetto.

— Ti presenterò la persona dell'assistenza che segue questo cliente. Potrà organizzare la conversazione.

Il mercoledì Lucas scoprì che dietro l'espressione «gestire i lotti» c'era una situazione molto meno semplice.

Il responsabile della qualità del cliente mostrò una lettera ricevuta da un fornitore. Una materia prima doveva essere messa da parte. Il fornitore conosceva il numero del lotto. Il produttore sapeva quali ordini aveva ricevuto. Ma per individuare esattamente i prodotti finiti interessati, i suoi team dovevano incrociare diversi documenti.

— L'ultima volta — spiegò il responsabile della qualità — eravamo in quattro a controllare file per buona parte della giornata. E dopo abbiamo comunque dovuto tornare in reparto per essere sicuri.

Lucas chiese di vedere come venivano registrate le informazioni durante la produzione. Il cliente mostrò una schermata dell'ERP e poi un foglio di calcolo.

— Ma perché annotate questo numero anche in Excel?

— Perché il vostro software non ci permette di trovare le informazioni che ci servono.

La risposta non era aggressiva. Proprio per questo era ancora più difficile da sentire.

Alla fine della chiamata, Lucas aveva tre pagine di appunti e meno certezze di quando aveva cominciato.

— Allora? — chiese Alice quando lui la richiamò.

— Avevo iniziato a immaginare una schermata per registrare i numeri dei lotti.

— E adesso?

— Adesso penso che dobbiamo innanzitutto capire come registrano ciò che hanno effettivamente utilizzato. Altrimenti avremo una bella schermata, ma non sapremo ancora rispondere alla loro domanda.

Alice annuì. Avrebbe potuto spiegarglielo già lunedì. Ma non le dispiaceva che ci fosse arrivato da solo.

8

Il venerdì successivo Lucas le mostrò una prima versione su un database dimostrativo.

Si poteva associare un lotto di materia prima a una lavorazione e poi ritrovare i prodotti finiti ottenuti. Aveva aggiunto un piccolo esempio per seguire un lotto fino ai prodotti interessati da una segnalazione.

Alice fece qualche clic.

— Hai fatto questo in settimana?

— Sì. Ci sono ancora molte cose sbagliate o mancanti, ma almeno abbiamo qualcosa da mostrare al cliente.

— E se aggiungessimo anche la possibilità di seguire il percorso al contrario?

— C'è già.

Aprì un'altra schermata.

Alice scoppiò a ridere.

— Naturalmente.

Dopo la dimostrazione del modulo ferie, il suo modo di valutare la portata dei progetti era cambiato. Ciò che prima avrebbe rinviato al trimestre successivo le sembrava ormai a portata di mano.

— Potremmo anche cercare per fornitore. E mostrare un avviso direttamente sugli ordini coinvolti. E perché non preparare i documenti per il richiamo dei prodotti?

Lucas non rispose subito.

— Possiamo annotare queste idee — disse infine. — Però prima vorrei tornare dal cliente con quello che abbiamo.

— Non vorrai mica aspettare quindici giorni per mostrargli una versione più completa?

— Non è la dimostrazione che mi preoccupa.

— Allora che cosa ti preoccupa?

Questa volta Lucas condivise una schermata molto meno piacevole da guardare. Tabelle vecchie, campi dai nomi incomprensibili e diversi collegamenti tra produzione e magazzino.

— Possiamo creare tutte le schermate che vogliamo. Il problema è che il nostro ERP non è mai stato progettato per conservare tutta la storia dei lotti. Ci sono movimenti di magazzino vecchi di anni, alcuni corretti manualmente. E certe informazioni semplicemente non esistono.

— L'IA può aiutarci a rimettere tutto in ordine, no?

— Può aiutarmi a scrivere una migrazione. Non può inventare ciò che non abbiamo mai registrato.

Alice rimase in silenzio.

— Se vogliamo farlo bene — continuò Lucas — dovremo modificare parte del modello dati e controllare gli effetti sugli altri moduli. E non possiamo permetterci di alterare le giacenze di un cliente per guadagnare due settimane.

— Non ti sto chiedendo di rompere il software.

— Lo so. Ma da venerdì, ogni volta che ti mostro qualcosa, me ne chiedi altre tre.

L'osservazione non aveva nulla di aggressivo. Ma Alice la sentì lo stesso.

— È vero — ammise.

— Anch'io ho voglia di andare veloce. Mi conosci. Però poter generare codice in un'ora non rende semplici quindici anni di funzionamento dell'ERP.

Dopo un attimo di esitazione, aggiunse:

— Thomas ha guardato una parte del codice del modulo ferie. Ha trovato due casi che avevamo gestito male. Aveva ragione. Li abbiamo corretti, ma mi ha ricordato una cosa: possiamo costruire molto più velocemente e avere comunque bisogno di qualcuno che conosca le insidie.

Alice ricordò Thomas il venerdì della dimostrazione, quando aveva chiesto soltanto di vedere il codice.

— Allora che cosa proponi?

— Continuiamo la dimostrazione con dati fittizi. Verifichiamo di aver capito bene le esigenze. Ma prima di toccare il magazzino reale, voglio che ci prendiamo il tempo di capire ciò che esiste già e fare i controlli necessari.

Alice guardò la prima versione, ancora aperta sull'altro schermo.

La settimana precedente aveva difeso l'autonomia di Lucas davanti a Hubert. All'improvviso le piaceva molto meno scoprire che quell'autonomia comprendeva il diritto di dirle di no.

— D'accordo — rispose infine. — Per adesso non tocchiamo i dati reali.

Il lunedì seguente Hubert cominciò la riunione settimanale con una domanda ormai familiare.

— Allora, il nostro esperimento?

— Sta andando avanti — rispose Alice. — Più velocemente del previsto su alcuni punti. Più lentamente su altri.

— Per esempio?

Alice gli mostrò la dimostrazione di Lucas e riassunse la discussione di venerdì.

Hubert si soffermò sulle riserve tecniche.

— Fa bene a stare attento — disse. — Non voglio che risolviamo un problema di tracciabilità creando problemi di magazzino a tutti i clienti.

— Sì. E credo di avergli messo un po' di pressione.

— Tu?

— Da due settimane ho la sensazione che tutto sia diventato possibile. Appena spunta un'idea, penso che possiamo provarla subito.

Hubert sorrise.

— Era esattamente come mi sentivo dopo la sua prima dimostrazione.

— Il problema è che è facile confondere la velocità con cui vediamo qualcosa funzionare sullo schermo con quella necessaria a ottenere qualcosa di affidabile.

— Almeno Lucas sa quando deve rallentare.

— Sì. E devo fidarmi del suo giudizio.

Due giorni dopo Julie Morel chiese ad Alice se aveva cinque minuti.

Alice conosceva bene quella formula. Con Julie, cinque minuti potevano facilmente diventare mezz'ora.

— La settimana scorsa ho incontrato un potenziale cliente piuttosto importante — esordì Julie. — Ha diversi stabilimenti e cerca un ERP che si adatti ai suoi modi di lavorare. Guarda cosa ho preparato.

Condivise una serie di prototipi grafici molto curati. Mostravano passaggi di approvazione differenti secondo la categoria d'ordine, regole specifiche per ogni stabilimento e schermate che cambiavano in base all'utente.

— Hai fatto tutto tu? — chiese Alice.

— Con ChatGPT. Ho ripreso gli appunti dell'incontro, gli ho chiesto di proporre un percorso e poi l'abbiamo corretto insieme. Ci ho passato una serata.

Alice colse subito il valore di quelle immagini. In poche ore Julie aveva dato forma concreta a un'esigenza che fino a quel momento esisteva soltanto in una relazione commerciale.

— E il cliente ha visto i prototipi?

— Non ancora. Volevo mostrarli prima a te. Ma mi ha detto chiaramente che, se fossimo stati capaci di fare queste cose, la trattativa avrebbe potuto cambiare.

Alice scorse le schermate.

— L'idea è interessante. Dovresti parlarne direttamente con Lucas.

Julie la guardò sorpresa.

— Direttamente? Pensavo che avresti preparato tu il progetto.

— Perché? Conosci il cliente, ci hai parlato, hai perfino preparato i prototipi. Puoi spiegargli ciò che cerca meglio di me.

— E Lucas può vedere se è fattibile?

— Sì. Ma Julie, attenzione: verificare la fattibilità non significa impegnarci a svilupparlo.

— Certamente.

Alice conosceva quel «certamente». Non era una bugia. Era l'entusiasmo di una commerciale davanti a una possibile vendita.

Il giorno successivo incontrarono Lucas. Julie mostrò i prototipi, spiegò le richieste del potenziale cliente e rispose alle domande.

Lucas si interessò immediatamente alle differenze tra le regole dei vari stabilimenti.

— Nell'ERP abbiamo già una parte di questo meccanismo — disse. — Non per fare esattamente questo, ma esiste. Probabilmente potremmo preparare qualcosa per una prova.

— In quanto tempo? — chiese Julie.

— Per una dimostrazione? Non necessariamente molto.

Julie guardò Alice. Il suo sorriso era quasi quello di chi aveva già concluso la vendita.

— Se glielo mostriamo la settimana prossima, potrebbe fare davvero la differenza.

Alice sentì la propria voglia di procedere rispondere prima ancora di aver riflettuto. Pensò al progetto di tracciabilità, alle richieste in sospeso, alle regole particolari di quel potenziale cliente e a tutti gli altri che forse non ne avrebbero mai avuto bisogno.

— No — disse.

Gli altri due si voltarono verso di lei.

— No a cosa? — chiese Julie.

— No all'idea di cominciare a sviluppare questa funzionalità nel nostro prodotto per ottenere quel contratto.

— Ma hai appena detto che è interessante.

— Lo è. Ma non sappiamo se questo cliente abbia bisogno di una funzione utile a tutti oppure di un modo di lavorare molto particolare dei suoi stabilimenti.

— Rischiamo di perdere la vendita.

— È possibile.

Julie non si aspettava evidentemente quella risposta.

— Finalmente abbiamo gli strumenti per rispondere in fretta a richieste come questa — insistette. — Sarebbe un peccato non approfittarne.

— Non voglio rinunciarci. Voglio che capiamo che cosa stiamo proponendo. Se aggiungiamo ogni richiesta particolare al cuore dell'ERP, finiremo con un programma impossibile da far evolvere.

— Ma Lucas dice che possiamo farlo.

— Lucas dice che può preparare una dimostrazione. Non è la stessa cosa.

Seguì un silenzio. Julie guardò un'ultima volta i suoi prototipi.

— E allora che cosa gli dico?

— Che abbiamo capito il problema e vogliamo mostrargli una possibilità. Non che abbiamo deciso di aggiungere la funzionalità al prodotto.

Julie annuì. Sembrava delusa, ma non scoraggiata.

— Va bene. Posso organizzare un incontro per farci spiegare meglio le differenze tra i suoi stabilimenti.

— Perfetto. E conserva i prototipi. Ci aiuteranno a fare le domande giuste.

Quando Julie lasciò la chiamata, Alice rimase qualche secondo con Lucas.

— Avresti davvero potuto costruirlo in fretta?

— Una prima versione, sì.

— E ti sarebbe piaciuto farlo?

— Sinceramente? Sì. Sembrava divertente.

Alice rise.

— Anch'io avevo voglia di dire sì.

— Che cosa ti ha fermata?

— Il fatto che stavamo parlando tutti e due di ciò che potevamo fare, e quasi per niente di ciò che dovevamo fare.

La settimana dopo Lucas mandò ad Alice un link con un messaggio.

Ieri ho fatto qualche prova. Puoi dare un'occhiata?

Alice aprì la pagina. Riconobbe l'interfaccia del loro ERP, ma c'era un piccolo pannello laterale che permetteva di modificare alcune regole.

Poteva scegliere uno stabilimento, cambiare una soglia di approvazione, mostrare o nascondere una fase e riavviare la dimostrazione.

Richiamò Lucas.

— Che cos'è?

— Un piccolo strumento per preparare dimostrazioni con regole diverse. Ho riutilizzato il nostro vecchio meccanismo di configurazione. Funziona soltanto in un ambiente di test.

— Quanto ci hai messo?

— Due ore, forse. Mi è venuta l'idea ripensando ai prototipi di Julie.

— Ma non è nemmeno il progetto a cui stai lavorando!

— No. Volevo vedere se potevamo evitare di modificare il codice ogni volta che volevamo provare una variante.

Alice iniziò a cambiare i parametri. Lo schermo rispondeva. Provò un'altra configurazione, poi una terza.

— Aspetta. Vuoi dire che posso preparare da sola diverse versioni di una dimostrazione?

— Con le regole che abbiamo messo a disposizione, sì. Non qualsiasi cosa. E soprattutto mai negli ambienti dei clienti.

— Ma posso mostrare a Julie due percorsi diversi e chiederle quale risponde davvero alle esigenze del suo potenziale cliente.

— Esatto.

Alice rimase a guardare la pagina per qualche istante. Il primo modulo ferie aveva reso Lucas più veloce. Questo strumento le offriva una possibilità che il giorno prima non aveva.

Poteva provare un'idea senza chiedere a uno sviluppatore di implementarla apposta per lei.

— Ti immagini che cosa potremmo fare?

Lucas sorrise.

— Credo di sì. Però prima che tu mi chieda di trasformarlo in un prodotto completo, ti ricordo che l'ho messo insieme in due ore.

— Promesso.

Chiuse la chiamata e riprese a giocare con i parametri.

Il lunedì successivo Hubert notò che Alice non aveva aperto il solito documento di monitoraggio.

— Non hai preparato nulla per oggi?

— Sì. Però voglio mostrarti una cosa.

Gli presentò lo strumento di Lucas e creò davanti a lui due varianti della stessa dimostrazione.

Hubert capì piuttosto in fretta.

— Non hai chiesto agli sviluppatori di fare queste modifiche?

— No. Lucas ha creato un piccolo strumento. Posso provare da sola certe regole.

— Potremmo lasciarlo usare ai commerciali?

— Forse più avanti. Per ora non è abbastanza solido. Ma Julie può già lavorare con me sui casi del suo potenziale cliente.

Hubert guardò lo schermo con interesse.

— E il progetto dei lotti?

— Lucas sta avanzando. È tornato dal cliente con la prima versione. Capiamo meglio che cosa bisogna fare. Però non vuole toccare i dati storici finché non avremo reso sicuro il modo in cui li gestiamo.

— Mi sembra ragionevole.

— Sì. Venerdì mi aveva irritata. Oggi sono contenta che abbia insistito.

Alice fece girare la penna tra le dita.

— Sta succedendo qualcosa di strano, Hubert. Lucas parla con i clienti e prende decisioni sulla soluzione. Julie arriva con prototipi quasi utilizzabili. E adesso io posso provare alcune varianti del nostro ERP senza passare dallo sviluppo.

— Ti preoccupa?

— A volte. Mi chiedo che cosa diventeranno le nostre professioni se tutti cominceranno a fare una parte del lavoro degli altri.

— E hai trovato una risposta?

— Non proprio.

Rifletté.

— So soltanto che non voglio essere la persona attraverso cui deve passare ogni idea. Julie non aveva motivo di aspettare che trasformassi i suoi prototipi in ticket. Lucas non ha bisogno che gli spieghi come costruire ogni schermata.

— Però hai rifiutato di sviluppare la richiesta del potenziale cliente.

— Sì. Perché quella era una decisione sul nostro prodotto. Potevo sbagliarmi, posso ancora cambiare idea, ma spettava a me assumermene la responsabilità.

— E Lucas ha rifiutato una scorciatoia sulle giacenze.

— Perché sa che cosa rischiamo di rompere. Su questo ha un giudizio che devo ascoltare, anche quando ho fretta.

Hubert osservò per un istante la piccola finestra in cui appariva Alice.

— Quindi possiamo permettere alle persone di fare più cose, ma non possiamo lasciare che le decisioni rimangano sospese tra loro senza un responsabile.

— È ciò che comincio a pensare.

Esitò, poi aggiunse:

— Prima, a volte, mi sembrava che il mio lavoro fosse importante perché molti progetti passavano attraverso di me. Adesso mi chiedo se non sia più importante ciò che otteniamo alla fine. Stiamo risolvendo il problema giusto? Stiamo costruendo qualcosa di affidabile? Vale davvero la pena farlo?

Hubert annuì.

— E se nessuno sa chi deve rispondere a queste domande, avremo un problema.

— Sì.

Rimasero in silenzio per qualche secondo. Nessuno dei due voleva trasformare la conversazione nell'annuncio di una nuova organizzazione.

Il loro esperimento era appena cominciato. Aveva aperto nuove possibilità, ma anche aggiunto molte decisioni da prendere.

Alice ricevette una notifica. Un messaggio di Julie: il potenziale cliente era disposto a vedere le varianti della dimostrazione. Voleva anche sapere quando sarebbero state disponibili per i suoi team.

Alice non rispose subito.

Il software permetteva ormai di produrre risposte molto più rapidamente.

Ma qualcuno doveva ancora decidere quali Malorean Corp. fosse pronta a promettere — e ad assumersi.

6 – I confini tra i ruoli si sfumano, le responsabilità restano

Per molto tempo, i confini tra Prodotto, Design e Ingegneria hanno offerto punti di riferimento semplici.

Il Prodotto comprendeva il problema e decideva che cosa costruire. Il Design progettava l'esperienza. L'Ingegneria trasformava quell'intenzione in software. Nella realtà questa divisione non è mai stata così netta, ma ha comunque strutturato i team, le assunzioni, i processi e le responsabilità.

La ripartizione era già messa in discussione prima dell'IA: i team di Prodotto autonomi descritti nel capitolo 1.4 facevano collaborare Prodotto, Design e Ingegneria su un problema comune. Ora l'IA accelera il superamento di un altro confine: quello dell'esecuzione, quando ciascuno può svolgere più attività un tempo riservate a una singola professione.

La panoramica delle organizzazioni ha mostrato questa evoluzione, in particolare attraverso lo studio di Figma. [1] Su Malorean ho descritto anch'io il movimento in entrambe le direzioni: gli sviluppatori si avvicinano alla comprensione del problema e i PM alla sua realizzazione. [2]

La tentazione è concludere che i ruoli scompariranno. Credo invece che questo cambiamento ci inviti a riformulare la responsabilità di ciascuno: **più i ruoli diventano permeabili, più le responsabilità devono essere esplicite.**

La capacità di agire non appartiene più a una sola professione

La prima cosa che l'IA ridistribuisce non è necessariamente il potere di decidere. È la capacità di agire.

Prendiamo i designer che modificano direttamente il codice, una pratica documentata da Atlassian: un team tecnico continua a verificare la qualità e le condizioni per il rilascio in produzione. [3] Questa fase non è un ritorno alla vecchia divisione del lavoro; dimostra che poter intervenire sul codice e rispondere della sua affidabilità sono cose diverse.

La stessa distinzione compare nell'offerta di lavoro di Linear per un Product Engineer: l'azienda si aspetta che questa persona segua un progetto dal problema iniziale al risultato consegnato. [4] Questo non le attribuisce automaticamente l'autorità su tutte le decisioni strategiche di Prodotto.

Possiamo quindi ridurre i passaggi di consegne senza eliminare le competenze che ne giustificavano alcuni. La domanda non è più quante professioni intervengano, ma **chi risponde di che cosa quando diverse professioni agiscono nello stesso ambito**.

Poter contribuire non significa essere responsabili di un ambito

Bisogna distinguere due cose che le organizzazioni spesso confondono: il permesso di agire e la responsabilità.

Un commerciale può costruire uno strumento interno. Questo non lo rende responsabile della sicurezza della piattaforma.

Un PM può modificare un'interfaccia. Questo non significa che diventi responsabile della coerenza del Design System.

Uno sviluppatore può decidere un comportamento locale del prodotto. Questo non gli attribuisce automaticamente la responsabilità della strategia, del posizionamento o della roadmap globale.

La distinzione esiste già in molti team, ma l'IA ci obbliga a renderla esplicita.

In un'organizzazione molto specializzata, professione e responsabilità tendevano a coincidere. Chi poteva agire apparteneva anche all'ambito di

cui rispondeva. Quando la capacità di agire si diffonde, questa coincidenza viene meno.

Il rischio è allora confondere autonomia e responsabilità diluita.

Ho già difeso un principio simile a proposito della qualità: gli sviluppatori non possono consegnare una modifica e poi considerare i difetti come un problema del team QA, di un turno dedicato ai bug o del prossimo sviluppatore disponibile. La qualità tecnica di ciò che viene consegnato resta una responsabilità identificabile. [5]

La permeabilità dei ruoli non cambia questo principio. Lo rende più importante.

Il concetto di Garante

Per rispondere a questo problema utilizzo il concetto di **Garante**.

Per ogni ambito importante, una persona o un team rimane garante della sua coerenza e qualità.

Il Garante non è necessariamente chi svolge tutto il lavoro di quell'ambito.

Non è neppure chi deve approvare ogni piccola decisione.

È la persona o il team a cui l'organizzazione si rivolge quando una scelta importante riguarda quell'ambito, quando si oltrepassa un limite o quando qualcosa va storto.

In un'organizzazione semplice, possiamo immaginare almeno tre garanzie esplicite:

- **Garante del Prodotto**: coerenza del prodotto, decisioni strutturali, clienti di riferimento, proposta di valore, grandi scelte, rapporto con la strategia.
- **Garante del Design**: Design System, coerenza dell'esperienza, accessibilità, nuovi pattern e livello di qualità dell'esperienza utente.

- **Garante dello Sviluppo**: architettura, sicurezza, prestazioni, manutenibilità, qualità tecnica e condizioni per la messa in produzione.

Il principio è semplice:

tutti possono contribuire a più ambiti, ma per ciascuno deve esserci qualcuno che ne risponde.

Questo modello permette di superare i confini professionali senza far sparire le responsabilità.

Un commerciale può costruire, senza diventare garante tecnico

Prendiamo un esempio volutamente semplice.

Un commerciale individua un'esigenza, usa uno strumento assistito dall'IA e crea una piccola applicazione o una modifica utile. Purché l'azienda disponga di un ambiente sufficientemente sicuro, non c'è motivo di sostenere che soltanto l'Ingegneria abbia il diritto materiale di costruire.

Questa libertà, tuttavia, non significa che la qualità tecnica diventi responsabilità del commerciale.

Se la modifica riguarda la sicurezza, i permessi, dati sensibili o l'architettura, il Garante dello Sviluppo deve poter chiedere una revisione, imporre una correzione o bloccare la modifica.

Il commerciale continua a rispondere dell'esigenza che vuole soddisfare.

L'Ingegneria resta garante di ciò che l'azienda accetta di mantenere dal punto di vista tecnico.

Lo stesso vale per il Design.

Un PM può comporre una schermata usando componenti conosciuti. Finché rimane nel Design System e utilizza pattern consolidati, chiedere sistematicamente una validazione al Design ridurrebbe inutilmente la sua autonomia.

Se invece esce dal Design System, inventa un nuovo pattern o progetta un'interazione importante e complessa, il Garante del Design deve poter intervenire.

Il confine non è più: «chi ha il diritto di toccare che cosa?».

Diventa: **«a partire da quale livello di rischio o novità deve intervenire il garante di un ambito?»**

Il responsabile di un progetto e il Garante di un ambito sono persone diverse

Questa è probabilmente una delle distinzioni più importanti delle nuove organizzazioni.

L'**owner** è responsabile di un progetto.

Deve comprenderne il problema, far avanzare il lavoro, prendere le decisioni locali, coordinare ciò che serve, verificare il risultato e rispondere del raggiungimento dell'obiettivo.

Il **Garante** assume una responsabilità più trasversale.

Non è necessariamente responsabile del progetto. Risponde di un determinato livello di coerenza, qualità o rischio nell'ambito di sua competenza.

Un Product Engineer può quindi essere owner di una funzionalità senza diventare garante dell'intera architettura aziendale.

Un PM può essere owner di un lancio senza diventare garante di tutto il Design.

Un commerciale può essere owner di uno strumento aziendale senza diventare garante della sicurezza delle applicazioni interne.

La distinzione rende più chiara la responsabilità sul risultato, senza pretendere che una sola persona sia esperta in ogni ambito.

AMPECO ne offre un esempio: un Product Owner e un ingegnere si allineano sul «che cosa» e sul «perché», poi gli agenti svolgono gran parte dell'analisi, dello sviluppo e dei test. L'esecuzione è distribuita, ma la validazione finale rimane una responsabilità umana. [6]

L'IA può dunque ridurre il numero di persone necessarie per eseguire un progetto e, allo stesso tempo, aumentare la necessità di sapere chiaramente **chi ne risponde**.

Forse il ruolo non è più l'unità migliore per leggere l'organizzazione

Questa evoluzione solleva una domanda più ampia: ha ancora senso organizzare l'azienda principalmente in base ai titoli professionali?

Il dibattito sul Product Owner è significativo. Su Malorean ho sostenuto che il PO sia prima di tutto una funzione — un mandato decisionale e di responsabilità — anziché una professione giustificata dalla sola gestione di un backlog. [7]

Con l'IA la distinzione diventa particolarmente utile.

Se definiamo un ruolo attraverso l'elenco dei compiti che svolge, l'IA rende instabile questa definizione. Molti compiti diventano accessibili a profili più diversi.

Se invece definiamo una responsabilità attraverso ciò di cui una persona o un team devono rispondere, la definizione resiste molto meglio.

Un PM può scrivere meno specifiche senza smettere di rispondere della qualità delle scelte di Prodotto.

Uno sviluppatore può scrivere meno codice manualmente senza smettere di rispondere della qualità tecnica.

Un designer può non essere più l'unico a spostare elementi sullo schermo senza smettere di essere garante della coerenza dell'esperienza.

Anche in Cursor, dove i confini tra Prodotto, Design e Sviluppo sono meno rigidi, il responsabile del Prodotto ricorda ciò che conta: scegliere cosa merita di essere costruito e dare un orientamento sufficiente a evitare deviazioni. [8]

Le professioni, quindi, cambiano non tanto perché perdono tutte le responsabilità, quanto perché non possiedono più il monopolio su alcune attività esecutive.

Il Garante non deve diventare una nuova burocrazia

Esiste però una trappola evidente.

Se ogni intervento di un PM sul codice deve essere approvato dall'Ingegneria, ogni decisione locale di uno sviluppatore deve passare dal PM e ogni schermata prodotta da chi non è designer deve aspettare un comitato di Design, allora non abbiamo risolto nulla.

Abbiamo semplicemente ricreato i vecchi passaggi obbligati con un altro nome.

Il Garante non deve diventare un'approvazione sistematica.

La sua presenza serve proprio a consentire più autonomia: ciascuno può agire dentro regole conosciute, senza chiedere continuamente il permesso, perché l'organizzazione sa dove risiedono le responsabilità e in quali casi i garanti devono tornare a intervenire.

Nella vita reale, d'altra parte, questo modello non funziona mai alla perfezione.

Arrivano in produzione cose che il Garante non avrebbe approvato esattamente in quel modo. Alcune decisioni locali sfuggono ai controlli. A volte i team si prendono più libertà del previsto. Altre persone, al contrario, chiedono un intervento troppo presto per prudenza.

Non significa necessariamente che il modello abbia fallito.

Un'organizzazione davvero autonoma accetta che alcune decisioni vengano prese senza ispezione preventiva. Altrimenti non è autonomia.

Il problema, quindi, non è garantire che non ci siano mai scostamenti.

È definire **quali scostamenti siano accettabili, quali il sistema debba rilevare e quali richiedano l'intervento di un Garante prima di diventare costosi**.

A questo punto la domanda cambia.

Abbiamo distinto due responsabilità: quella di chi conduce un progetto e quella del Garante di un ambito.

Resta da risolvere un'altra questione:

quando dobbiamo lasciare che l'owner decida da solo e quando deve intervenire il Garante?

In Malorean Corp. questo confine tra iniziativa e responsabilità comincia a creare attriti. Li ritroveremo nel prossimo episodio, prima di esaminare le misure di protezione e le regole per richiedere l'intervento dei responsabili.

Bibliografia

[1] Are Roles and Responsibilities a Thing of the Past? – Figma

- URL: https://www.figma.com/blog/2025-shifting-roles-report/
- Studia l'ampliamento delle responsabilità e la maggiore permeabilità delle professioni nei moderni team di sviluppo prodotto.

[2] PM, développeurs et IA : les rôles se brouillent, les responsabilités restent – Malorean

- URL: https://malorean.net/articles/2026-08-01-pm-developpeurs-et-ia-les-roles-se-brouillent-les-responsabilites-restent.it.html

- Illustra la lettura dell'autore: l'IA amplia chi può agire senza eliminare le responsabilità di Prodotto, Design e Ingegneria.

[3] Designers' workflow for shipping code – Atlassian

- URL: https://www.atlassian.com/blog/ai-at-work/designers-workflow-for-shipping-code
- Descrive designer che modificano direttamente il codice del prodotto, con la revisione dell'Ingegneria mantenuta per qualità e rilascio.

[4] Product Engineer – Linear Careers

- URL: https://linear.app/careers/0c7c2e26-0a98-42cf-a47c-9a3999fb513b
- Offerta di lavoro sulle responsabilità richieste da Linear a un Product Engineer, dalla definizione del problema alla consegna.

[5] La qualité appartient à ceux qui livrent – Malorean

- URL: https://malorean.net/articles/2026-06-03-la-qualite-appartient-a-ceux-qui-livrent.it.html
- Sostiene una responsabilità tecnica esplicita: dopo la consegna, la qualità non può essere delegata a chi deve riparare gli errori.

[6] How we built an AI-native engineering system – AMPECO

- URL: https://www.ampeco.com/blog/how-we-built-an-ai-native-engineering-system/
- Mostra un'esecuzione ampiamente delegata agli agenti, mantenendo intenzioni, architettura e validazione sotto responsabilità umana.

[7] Le PO n'est pas un métier, c'est une fonction – et l'IA n'y change rien, elle accélère – Malorean

- URL: https://malorean.net/articles/2026-08-03-le-po-nest-pas-un-metier-cest-une-fonction.it.html

- Distingue un mandato decisionale e di responsabilità da una professione definita soprattutto dalla gestione di documenti o backlog.

[8] How Cursor Is Navigating the New Shape of Product Craft – Atlassian / Cursor

- URL: https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Mostra confini professionali più permeabili, mantenendo giudizio di Prodotto, obiettivi e controlli come responsabilità centrali.

7 – Malorean Corp. – Ciò che Alice sapeva senza averlo mai scritto

Il giovedì mattina Lucas chiamò Alice per la terza volta quella settimana.

— Ti rubo solo due minuti — promise.

Alice aveva imparato a diffidare di quella frase. Lucas non le chiedeva mai riunioni lunghe. Arrivava con una domanda precisa, ne discutevano e tornava a lavorare. Proprio per questo era difficile rifiutare quelle interruzioni.

— Dimmi.

— Sulla tracciabilità dei lotti, il cliente vorrebbe cercare anche per fornitore. Posso aggiungerlo alla dimostrazione. Però, se il fornitore ha consegnato più lotti con lo stesso numero interno, li mostriamo insieme o gli chiediamo di scegliere?

Alice rifletté qualche secondo.

— Soprattutto non dobbiamo dare l'impressione di aver identificato un lotto se ce ne sono diversi che corrispondono. Mostra i risultati separatamente. Poi vedremo con il cliente come distingue i lotti.

— Era quello che pensavo. Ma non ero sicuro.

— E allora perché mi chiami?

Lucas rise, un po' imbarazzato.

— Dopo la storia dei prototipi di Julie preferisco chiederti prima di prendere una decisione che assomigli a una regola di Prodotto.

La settimana prima Alice gli aveva chiesto di non trasformare la richiesta di un potenziale cliente in una funzionalità standard. Ora le dispiaceva che ne avesse ricavato l'idea di doverla consultare al minimo dubbio.

— Va bene. Ma per questa decisione non avevi bisogno di me.

— Mi rassicura sentirtelo dire.

Chiusero la chiamata. Erano passati sette minuti. Alice tornò al documento che stava scrivendo, rilesse l'ultima frase e non ricordò più che cosa volesse aggiungere.

Dieci minuti dopo Lucas le mandò un messaggio.

Un'altra domanda: per le vecchie produzioni di cui non abbiamo i dati sui lotti, scriviamo «sconosciuto» o nascondiamo la colonna?

Alice iniziò a rispondere. Poi si fermò.

La domanda era diversa, ma il ragionamento era lo stesso: non presentare mai come certa un'informazione che l'ERP non possiede. Avrebbe potuto comunicargli quel principio invece di decidere separatamente i due casi.

Rispose comunque.

Mostriamo chiaramente che manca l'informazione. Non nascondiamo il problema.

Lucas reagì con un pollice alzato.

Alice guardò l'agenda. Una conversazione con l'assistenza era stata rimandata. Nel pomeriggio aveva un incontro con un cliente. Tra i due doveva rivedere le proposte di Julie per il potenziale cliente con più stabilimenti.

Le piaceva che Lucas le parlasse direttamente. Era proprio ciò che aveva voluto affidandogli il progetto. Ma avanzava così in fretta che a ogni nuova possibilità corrispondeva una nuova domanda. E più la consultava, meno

tempo Alice aveva per fare ciò per cui era più preparata: capire i clienti e riflettere sulla direzione del prodotto.

Anche Lucas perdeva tempo. A volte aspettava la sua risposta per riprendere a lavorare.

Le passò per la testa un pensiero spiacevole: se continuava così, forse avrebbero finito per rimettere tutte le decisioni nei ticket. Sarebbe stato più semplice organizzarsi. Lei avrebbe scritto le specifiche, Lucas le avrebbe eseguite e nessuno avrebbe più dovuto chiedersi quanto si estendesse la sua autonomia.

Esattamente ciò che avevano cercato di superare.

A mezzogiorno Alice si collegò con Bob in videoconferenza. Avevano preso l'abitudine di parlare senza ordine del giorno, per confrontare le esperienze.

— Come va con Lucas? — chiese Bob.

— Troppo bene, in un certo senso.

— Non sembra un complimento.

— Va velocissimo, si accorge di un sacco di cose, parla con i clienti. Però mi chiama continuamente per decisioni che potrebbe prendere da solo.

— E tu gli rispondi?

— Ovviamente.

— Allora fa bene a chiamarti.

Alice alzò gli occhi al cielo.

— Grazie, Bob.

— Sto scherzando solo a metà. Se posso avere una tua risposta in cinque minuti invece di rischiare di sbagliare, perché dovrei rinunciare?

— Perché così non riusciremo mai a lavorare.

Bob le chiese un esempio. Alice raccontò le due domande del mattino, poi la discussione con Julie e infine un dubbio recente di Lucas: dovevano aggiungere un'opzione richiesta da un cliente se probabilmente sarebbe servita soltanto a lui?

— Ti rendi conto? — disse Alice. — Sono sempre varianti della stessa domanda.

— E qual è la tua risposta?

— Dipende.

Bob sorrise.

— Ecco. È proprio il genere di risposta che obbliga le persone a richiamarti.

Alice stava per protestare. Ma aveva ragione.

— Non è un «dipende» casuale — continuò. — Guardo se rispondiamo a un problema ricorrente, se è utile ai nostri clienti prioritari, se complicherà il prodotto per tutti, se riusciremo a mantenere la soluzione…

Si interruppe.

— Vedi? — disse Bob. — Mi hai appena dato una parte della regola.

— Non è una regola. È più un modo di ragionare.

— Ancora meglio. Forse è questo che dovresti trasmettere a Lucas.

Alice rimase in silenzio.

Da anni scriveva ticket. Descriveva i comportamenti attesi, i casi particolari, i criteri di accettazione. Ma quasi mai metteva per iscritto ciò che le permetteva di scegliere tra diversi comportamenti possibili. Quel ragionamento rimaneva nella sua testa o svaniva nel corso di una riunione, una volta consegnato il lavoro.

— Probabilmente ho centinaia di cose del genere — disse.

— Sicuramente. Anch'io.

— E non abbiamo mai avuto motivo di scriverle.

— I motivi c'erano. Non avevamo tempo. E nessuno apriva un ticket intitolato «spiegare perché a volte rifiutiamo una buona idea».

Risero entrambi.

Dopo la chiamata, Alice aprì un documento nuovo. Scrisse: *Principi per le decisioni di Prodotto*.

Aggiunse una prima frase, la cancellò, ne provò un'altra. Dopo quindici minuti aveva ottenuto un paragrafo che sembrava un regolamento aziendale. Era corretto. Ma non assomigliava per niente a ciò che aveva spiegato a Bob.

Chiuse il documento.

Poi aprì ChatGPT.

— Aiutami a tirare fuori dalla testa quello che so del nostro prodotto — dettò. — Non voglio una procedura teorica. Fammi delle domande, aiutami a ricordare esempi e a distinguere le regole vere dalle mie semplici abitudini.

La prima risposta le suggerì di partire dalle decisioni recenti.

Alice cominciò a parlare.

Raccontò del potenziale cliente di Julie, dei prototipi accattivanti e del rischio di trasformare una necessità particolare in una promessa valida per tutti i clienti. Spiegò che un grande cliente poteva rappresentare molto denaro senza indicare automaticamente la direzione giusta per il prodotto.

— Aspetta, devo spiegarti perché — aggiunse ad alta voce. — Il nostro ERP è utilizzato da aziende molto diverse. Se aggiungiamo ogni regola particolare al cuore del software, finiamo per complicare la vita a tutti. Ma a volte una richiesta particolare rivela un problema molto più generale. Non possiamo neppure rifiutare tutto.

ChatGPT le chiese come facesse a distinguere i due casi.

Alice descrisse le conversazioni con l'assistenza, i casi osservati presso diversi clienti, le richieste dei commerciali e i vincoli delle implementazioni. Ricordò vecchie discussioni di cui non aveva conservato alcun verbale. Spiegò perché alcune opzioni erano state scartate, perché avevano preferito una configurazione a una nuova funzionalità e perché un'eccezione sulle paghe non si trattava come una preferenza d'interfaccia.

A volte si sorprendeva a rispondere a una domanda che non aveva mai pensato di porre esplicitamente.

L'IA riformulava, raggruppava e proponeva un principio. Alice lo correggeva.

— No, non è che «non sviluppiamo mai funzionalità specifiche». Possiamo farlo a certe condizioni, ma dobbiamo scegliere consapevolmente quanto ci costa. E soprattutto non dobbiamo far passare un'ipotesi commerciale per una decisione di Prodotto.

Più tardi, ChatGPT scrisse che una funzionalità di tracciabilità poteva garantire la ricostruzione di tutti i lotti storici.

— Assolutamente no — disse Alice davanti allo schermo. — Se i dati non sono mai stati registrati, non possiamo ricostruirli per magia. È un limite da mostrare, non una capacità da promettere.

Fece correggere il passaggio. Chiese di distinguere nel documento ciò che era stato deciso, ciò che riteneva probabile e ciò che andava verificato con

l'assistenza, Lucas o i clienti. Evitò di inserire informazioni riservate sugli account reali e riformulò gli esempi.

Passò un'ora. Poi due.

Il documento si riempiva di cose che Alice sapeva da tempo: i clienti ai quali Malorean voleva rivolgersi prioritariamente, i problemi ricorrenti, i compromessi accettati, gli impegni da non assumere, i limiti dei dati vecchi, le ragioni per cui un prototipo non era una funzionalità pronta per essere venduta.

Aggiunse collegamenti alle informazioni già disponibili: segnalazioni dell'assistenza, decisioni ancora recuperabili, appunti degli incontri, documentazione del prodotto. Dove mancava una prova, il testo doveva dirlo.

Verso la fine della terza ora ChatGPT propose una sintesi.

Alice la rilesse. Alcune pagine erano troppo lunghe. Altre mescolavano un principio generale e un esempio. Corresse un'ultima volta le formulazioni troppo categoriche, aggiunse una data e il perimetro del documento.

Poi esportò tutto in PDF.

Scorse le pagine, sorpresa da quante fossero.

Tre ore prima voleva soprattutto spiegare a Lucas quando poteva decidere senza di lei. Ora aveva tra le mani un ritratto imperfetto, ma sorprendentemente ricco, di ciò che il team sapeva del prodotto senza averlo mai condiviso per iscritto.

Non era una verità definitiva. Alcuni passaggi avrebbero richiesto conferme o aggiornamenti. Il PDF non avrebbe sostituito né i clienti, né i dati, né il giudizio dei responsabili.

Ma almeno la persona successiva non avrebbe dovuto ricominciare da zero.

Lucas aprì il file la mattina dopo.

— Hai scritto tutto questo per me?

— Non da sola — rispose Alice. — Ho passato tre ore a parlare con ChatGPT. Mi ha aiutata a formulare domande, organizzare risposte e individuare quello che mancava. Poi ho riletto e corretto. Ma la maggior parte di queste cose le sapevo già.

— Allora perché non le avevo mai viste?

Alice sorrise, un po' imbarazzata.

— Perché non entravano in nessun ticket.

Lucas sfogliò l'indice. Trovò principi, esempi di decisioni, vincoli del settore, informazioni sui clienti, link alle fonti e domande ancora aperte.

— Posso dare questo documento alla mia IA quando lavoro sulla tracciabilità?

— Sì, nel nostro ambiente autorizzato. Però dille di usarlo come contesto, non come prova che tutto sia vero. E se una regola è importante, torniamo alla fonte o alla persona che ne risponde.

— Mi farebbe risparmiare un sacco di tempo. A volte chiedo all'IA di aiutarmi a scegliere tra due soluzioni, ma conosce soltanto ciò che le ho raccontato. Può propormi una risposta molto convincente che non tiene conto di una decisione presa tre anni fa.

Alice conosceva bene la sensazione. Una risposta ragionevole, presa da sola, poteva essere assurda per il loro ERP.

Lucas si fermò su un passaggio.

— Qui dici che non dobbiamo mai nascondere un'informazione aziendale mancante, se questo può far credere che il risultato sia completo.

— È quello che ti ho detto ieri.

— Sì. Se avessi avuto questa frase ieri, non ti avrei chiamata.

Continuò a leggere. Poi sollevò la testa.

— D'accordo, ma ho ancora un problema.

Alice provò un attimo di scoraggiamento.

— Quale?

— Adesso capisco meglio come ragioni. Ma non so ancora quando ho il diritto di decidere da solo.

Lei posò il caffè.

Aveva appena individuato la lacuna più importante del documento.

Ripresero l'argomento insieme. Alice condivise una pagina bianca.

— Proviamo qualcosa di semplice. Prima di chiamarmi ti fai tre domande: ho capito il risultato che cerchiamo? La mia decisione rimane entro le regole che conosciamo? E che cosa potrebbe andare storto se sbagliassi?

— Dovremo definire che cosa significa «entro le regole».

— Appunto.

Scrissero tre livelli.

Verde: Lucas decideva da solo quando la scelta era locale, reversibile, conforme alle regole conosciute e non comportava una nuova promessa al cliente. Doveva essere in grado di spiegare il ragionamento e verificare il risultato.

Arancione: consultava Alice quando incontrava una situazione nuova, una contraddizione tra obiettivi o un dubbio significativo sul comportamento del prodotto. In presenza di un rischio tecnico coinvolgeva anche le persone competenti, come Thomas quando si trattava della storia dell'ERP.

Rosso: nessuno decideva da solo se bisognava modificare una regola strutturale, impegnare Malorean commercialmente, intervenire su dati sensibili o assumere un rischio importante e difficilmente reversibile. Occorreva una decisione esplicita insieme ai responsabili interessati e, talvolta, a Hubert.

Lucas rilesse la tabella.

— Quindi, se cambio la presentazione dei risultati nella dimostrazione per evitare confusione, sono nel verde.

— Sì.

— Se voglio aggiungere un comportamento che potrebbe riguardare tutti i nostri clienti, vengo da te.

— Ne discutiamo, sì. Soprattutto se non abbiamo capito bene l'esigenza.

— E se voglio modificare i dati storici per registrare i lotti?

— Non posso darti io da sola il via libera. Servono l'analisi tecnica e i controlli appropriati. Me l'hai ricordato abbastanza volte.

Lucas sorrise.

— Volevo essere certo che te ne ricordassi.

Aggiunsero una regola: l'assenza di una risposta da parte di Alice non avrebbe mai trasformato una decisione arancione o rossa in una verde. D'altra parte, non avrebbe dovuto chiederle il permesso per ogni decisione verde.

— E se sbaglio colore? — chiese Lucas.

— Me lo dici. Aggiustiamo le regole. Preferisco migliorare il sistema piuttosto che rimproverarti di non aver indovinato quello che pensavo.

Lucas annuì.

— Potrei perfino chiedere alla mia IA di confrontare le mie scelte con questa tabella, indicando le ragioni e le informazioni mancanti.

— Sì. Purché non decida al nostro posto se un rischio sia accettabile.

Alice aggiunse la tabella al documento di riferimento. Cominciava a capire la differenza tra trasmettere informazioni e dare davvero alle persone i mezzi per decidere.

Avrebbero potuto fermarsi lì. Ma Lucas tornò alla dimostrazione della tracciabilità.

— Parliamo della mia autonomia da un'ora. Ti faccio una domanda sul progetto stesso: come sapremo di averlo risolto bene?

Alice rispose quasi automaticamente:

— Se il cliente riesce a ritrovare i prodotti fabbricati con un determinato lotto.

— Sì, ma come lo verifichiamo? E quanto deve essere affidabile?

Questa volta Alice non cercò subito una risposta.

Avevano passato giorni a esplorare schermate, regole e strutture di dati. Avevano una dimostrazione promettente. Eppure nessuno aveva ancora scritto precisamente come giudicare il risultato.

Alice si ricordò del responsabile della qualità che aveva passato buona parte della giornata a incrociare file dopo una segnalazione del fornitore.

— Dovremmo partire da lì — disse. — Se riceve una segnalazione su un numero di lotto, può identificare le produzioni interessate senza rifare tutte le ricerche a mano? Quanto tempo ci mette? Il risultato è completo? E quando non abbiamo l'informazione, lo indichiamo chiaramente?

— Possiamo provarlo con lui. Non limitarci a chiedere se gli piace la dimostrazione.

— Esattamente.

Alice aprì una pagina nuova.

— In effetti avremmo dovuto cominciare da questo, prima di partire.

Lucas scoppiò a ridere.

— Se me l'avessi chiesto due settimane fa, avrei detto che cercavi di trasformare tutto di nuovo in un ticket.

— Forse. Ma guarda che cosa stiamo scrivendo.

Annotarono il problema osservato, gli utenti interessati, il risultato atteso, come misurarlo, i limiti dei dati, ciò che rimaneva fuori dal perimetro e le decisioni che richiedevano un'approvazione. Una pagina, non di più.

Lucas propose di chiamarla *Mini PRD*.

— Non mi stai dicendo quale schermata costruire — osservò. — Ci stiamo accordando su che cosa renderà utile il lavoro.

— E sui limiti che non dobbiamo oltrepassare.

Alice ragionò ad alta voce:

— Per i prossimi progetti dovrebbe essere il nostro punto di partenza. Prima di chiederci che cosa sviluppare, ci chiediamo quale risultato vogliamo ottenere e come lo verificheremo. Solo dopo definiamo il perimetro e decidiamo che cosa merita di essere costruito.

— E se so compilare questo documento con le informazioni giuste, posso cominciare senza aspettare che tu mi prepari un backlog.

— Puoi proporre come impostare il progetto. Non significa che approviamo qualsiasi impegno senza verificarlo. Ma sì, è questo l'obiettivo.

Completarono il Mini PRD sulla tracciabilità lasciando aperte le domande a cui non sapevano ancora rispondere. Rimanevano alcuni casi da verificare con il cliente e alcuni vincoli tecnici da confermare.

Il documento non dava una falsa impressione di certezza. Spiegava che cosa volevano imparare.

Alice lo trovò più utile di un elenco di funzionalità da spuntare.

Il lunedì successivo, l'incontro settimanale di Alice con Hubert fu diverso.

Alice non condivise né il solito quadro di monitoraggio né la lista delle funzionalità sviluppate da Lucas.

Gli mostrò la tabella delle decisioni e poi il Mini PRD.

Hubert lesse le prime righe.

— Avete passato tutto quel tempo a preparare documenti proprio ora che Lucas lavora molto più velocemente?

Alice conosceva l'obiezione. L'aveva fatta anche lei qualche giorno prima.

— Credo proprio che ci mancassero documenti di questo tipo. Scrivevamo che cosa costruire, ma raramente le informazioni necessarie a scegliere. Così Lucas mi chiedeva continuamente di decidere. Con questo può prendere alcune decisioni senza di me.

— E come sai che resta entro i limiti?

— Abbiamo definito quando può decidere da solo e quando deve consultarci. Per il resto, non controlleremo tutto. Verificheremo i risultati e correggeremo le regole se necessario.

Hubert si fermò sulla riga relativa ai dati vecchi.

— Non potrete garantire la tracciabilità di ciò che non è mai stato registrato.

— No — rispose Alice. — E il documento lo dice. Dobbiamo mostrarlo chiaramente al cliente.

Hubert annuì.

— Preferisco saperlo prima che un commerciale prometta il contrario.

— Appunto — disse Alice. — Il contesto non servirà soltanto a Lucas. Anche Julie potrà consultare ciò che sappiamo e ciò che non abbiamo ancora deciso. Però bisognerà mantenere aggiornate le informazioni e chiarire quali parti fanno autorità.

Concordarono di conservare una versione di lavoro modificabile, di cui il PDF sarebbe stato una fotografia datata. Le decisioni importanti sarebbero state registrate insieme alle ragioni; i principi ricorrenti avrebbero potuto, dove opportuno, diventare istruzioni per gli agenti o controlli negli strumenti. Nulla di questo avrebbe sostituito i test tecnici né le approvazioni necessarie prima della messa in produzione.

— Non voglio scoprire tra sei mesi che l'IA sta applicando una vecchia regola che ormai nessuno sostiene — disse Hubert.

— Nemmeno io — rispose Alice.

Per la prima volta Hubert sembrò interessarsi soprattutto al Mini PRD.

— E verificherete se il cliente riesce davvero a rintracciare i suoi lotti?

— Sì — rispose Alice. — Gli faremo provare dei casi insieme a noi. Se il risultato non è affidabile, non importerà quante schermate abbiamo finito.

Hubert chiuse il documento.

— Molto bene. Voglio sapere che cosa avete imparato, non solo che cosa avete consegnato.

Il mercoledì Alice notò qualcosa di strano nella sua messaggistica.

Nessun messaggio di Lucas dalla mattina.

Controllò il canale del progetto. Eppure c'era attività: Lucas aveva provato diversi modi di presentare i risultati nell'ambiente dimostrativo. Per uno di essi aveva spiegato di aver scelto di separare i lotti con riferimenti identici per non suggerire un collegamento inesistente. Citava il principio contenuto nel contesto di Prodotto, indicava i dati utilizzati e proponeva di verificare con il cliente se la schermata fosse comprensibile.

Non aveva chiesto il permesso.

Alice lesse le spiegazioni e tornò al lavoro. Non aveva nulla da aggiungere.

Nel tardo pomeriggio Lucas la chiamò.

— Ho un arancione che forse sta diventando rosso — disse.

— Ti ascolto.

— Il cliente vorrebbe cercare anche nelle vecchie produzioni. Per una parte della storia non abbiamo un collegamento affidabile tra materie prime e prodotti finiti. Posso mostrare ciò che sappiamo, ma se presentiamo una ricerca completa, potrebbero credere che non trovare risultati significhi che nessun prodotto è coinvolto.

Alice si raddrizzò.

— Hai ragione. Non possiamo lasciare quest'ambiguità.

— Potrei aggiungere un avvertimento e distinguere i periodi per i quali i dati sono utilizzabili. Però vorrei decidere insieme che cosa possiamo promettere al cliente.

— In questo caso sì. E dovremo chiedere anche all'assistenza come il cliente potrebbe interpretare quel limite.

Lucas condivise lo schermo. Cominciarono a lavorare sul problema. Non si trattava di approvare un pulsante né di dare una risposta che Alice

conoscesse già. Avevano davanti una decisione che poteva cambiare la fiducia del cliente nel software.

Durante la conversazione Alice si accorse di non aver pensato neppure una volta al tempo che Lucas le stava facendo perdere.

Prima di chiudere la chiamata, lui esitò.

— Sai, il tuo PDF non ha risposto a tutte le mie domande.

— Per fortuna — disse Alice.

— Per fortuna?

— Se fossi riuscita a metterci tutte le risposte, vorrebbe dire che non ti resterebbe più niente da scoprire.

Lucas sorrise.

— D'accordo. Ma credo di capire meglio quando devo rivolgermi a te.

Dopo la chiamata Alice riaprì l'agenda. Aveva un'ora libera prima dell'appuntamento successivo con un cliente.

Un'ora intera.

Non sapeva ancora se il loro esperimento avrebbe funzionato su scala più ampia, né quanti errori avrebbero commesso. Sapeva soltanto che Lucas aveva preso diverse decisioni senza di lei ed era tornato proprio quando il suo giudizio contava davvero.

Aggiunse al documento condiviso una nuova questione da chiarire sulla presentazione dei dati storici. Non un ticket che dicesse a Lucas quale codice scrivere. Una decisione da approfondire insieme alle persone interessate.

Il loro modo di lavorare non era diventato meno esigente.

Stavano semplicemente imparando a concentrare l'attenzione dove era davvero necessaria.

8 – Più autonomia richiede regole più chiare, non meno regole

Un'organizzazione può decidere che più persone debbano poter agire senza aspettare uno specialista.

La permeabilità dei ruoli e la distribuzione delle responsabilità esaminate fin qui creano un nuovo problema di coordinamento: più persone possono intervenire sullo stesso progetto senza avere necessariamente gli stessi riferimenti per decidere.

Ma se ogni nuova capacità di agire richiede un'approvazione umana sistematica, l'autonomia rimane teorica.

Abbiamo soltanto spostato i passaggi obbligati da una professione all'altra.

La vera sfida consiste quindi nel creare un sistema in cui più decisioni possano essere prese localmente senza perdere coerenza, sicurezza, qualità o direzione di Prodotto.

Qui emerge un paradosso importante:

più vogliamo distribuire l'autonomia, più le regole devono essere esplicite.

L'autonomia non nasce dall'assenza di regole.

Nasce dalla possibilità di agire senza chiedere continuamente il permesso, perché obiettivi, limiti, rischi e responsabilità sono abbastanza chiari.

Se tutto richiede un'approvazione, non c'è autonomia

Immaginiamo un Product Engineer a cui affidiamo la responsabilità di un progetto.

Parla con gli utenti, esplora una soluzione, crea prototipi, sviluppa e prepara il lancio.

Sulla carta è autonomo.

Ma se deve chiedere al PM di approvare ogni piccola decisione di Prodotto, al designer ogni dettaglio dell'interfaccia e al Tech Lead ogni scelta tecnica, l'organizzazione non ha davvero spostato le responsabilità. Ha soltanto costretto una persona ad attraversare più approvazioni.

Lo stesso ostacolo riguarda le altre professioni: permettere a qualcuno di costruire serve a poco se ogni gesto richiede i processi e le verifiche di prima.

La domanda quindi non è soltanto **chi può agire**.

Bisogna capire **entro quali limiti quella persona può decidere da sola**.

In Cursor questa logica è espressa attraverso obiettivi e meccanismi di controllo abbastanza chiari da permettere a chi costruisce di agire senza allontanarsi dalla visione o dalla roadmap. [1]

L'organizzazione sostituisce così una parte delle istruzioni dettagliate con un quadro per prendere decisioni.

Sostituire una parte dei ticket con criteri decisionali

Abbiamo visto, con Linear, perché un ticket che prescrive tutto può limitare l'iniziativa. [2] Adesso la questione diventa pratica: di quali informazioni deve disporre un team per decidere da solo senza aspettare un'altra specifica?

Questo non significa eliminare i ticket. Su Malorean ho sostenuto che il backlog è uno strumento per agire e coordinarsi, non la memoria completa del Prodotto. [3]

Un quadro di autonomia potrebbe quindi rendere esplicite le informazioni che orientano le decisioni locali senza fissare in anticipo ogni soluzione.

Può contenere:

- il cliente o l'ICP prioritario;
- il problema da risolvere;
- il risultato atteso;
- i limiti del perimetro;
- i vincoli importanti;
- il livello di qualità richiesto;
- ciò che l'organizzazione accetta esplicitamente di non fare;
- le soglie oltre le quali una decisione deve essere portata a un livello superiore.

Ramp offre un esempio concreto: chiarire gli ICP principali, i problemi prioritari e ciò che l'azienda vuole risolvere particolarmente bene permette anche di accettare il rifiuto di opportunità redditizie quando spingono il prodotto in un'altra direzione. [4]

Figma ricorda, da parte sua, che passare rapidamente da zero a uno non basta. I casi reali, le eccezioni e il livello di qualità rimangono determinanti per trasformare un prototipo rapido in un buon prodotto. [5]

Questo schema illustra una scelta organizzativa: non limitarsi a precisare **che cosa bisogna fare**, ma rendere visibili i limiti da rispettare. Non lo presento come un metodo universale.

Il contesto diventa un'infrastruttura

Per prendere buone decisioni a livello locale, bisogna poter accedere alle informazioni che consentono di comprendere il problema.

È probabilmente uno dei cambiamenti organizzativi più importanti dell'era degli agenti.

Un'organizzazione non può chiedere a più persone di decidere e, allo stesso tempo, conservare la maggior parte del contesto nella testa di pochi

specialisti. Linear sottolinea che i riscontri dei clienti, la direzione strategica, le decisioni e il codice devono essere accessibili tanto alle persone quanto agli agenti. [2]

Su Malorean ho descritto il ruolo del PM come quello di un **architetto del Contesto**: trasformare conoscenze disperse — interviste, obiezioni commerciali, assistenza, decisioni, dati, fonti e segnali deboli — in una risorsa organizzativa riutilizzabile. [6]

Questa idea cambia profondamente la funzione della documentazione.

L'obiettivo non è produrre più pagine.

È rendere accessibili gli elementi di cui qualcuno ha bisogno per decidere senza dover ricostruire ogni volta tutto il ragionamento.

Un buon contesto permette di comprendere:

- ciò che i clienti hanno realmente detto;
- quali ipotesi sono già state provate;
- quali vincoli contano;
- quali decisioni strutturali sono state prese;
- quali compromessi sono già stati accettati;
- quali metriche permettono di valutare il risultato.

Più cresce l'autonomia, più questo contesto deve essere affidabile e condiviso.

Altrimenti l'azienda distribuisce capacità di agire senza distribuire capacità di comprendere.

Conservare la memoria del perché

Questa ridistribuzione crea un altro problema.

Più decisioni vengono prese localmente, a volte da persone diverse, a volte con agenti che non erano presenti quando si erano affrontate le scelte precedenti.

Ma una decisione senza memoria torna presto a essere oggetto di discussione.

Si ricorda la conclusione, ma non più il contesto.

Si sa che esiste una regola, ma non più perché sia stata scelta.

Arriva un nuovo cliente, un nuovo manager assume l'incarico, cambia un team e una discussione già risolta ricomincia con meno informazioni della prima volta.

È precisamente il ruolo che attribuisco ai **Product Decision Record**. [7]

Un PDR non documenta ogni piccola decisione.

Registra quelle decisioni di Prodotto abbastanza importanti da orientare più team, più prodotti o diverse situazioni future.

Conserva, tra l'altro:

- il contesto;
- le alternative prese in considerazione;
- la decisione adottata;
- le ragioni della scelta;
- le conseguenze positive e negative;
- i rischi accettati;
- le condizioni che giustificherebbero una riapertura della decisione;
- le persone che hanno deciso o sono state consultate.

L'obiettivo non è aggiungere burocrazia.

Al contrario, un buon PDR deve evitare di ripetere cinquanta volte la stessa discussione.

Qui diventa utile una distinzione:

il codice conserva la traccia di ciò che è stato costruito; la memoria delle decisioni deve conservare il perché di certe regole strutturali.

L'autonomia duratura ha bisogno di entrambe.

Integrare le misure di protezione nel sistema

Anche con un buon contesto, sarebbe inefficiente chiedere a ogni dipendente di memorizzare tutte le regole dell'organizzazione.

Alcune aziende spostano così i controlli ripetitivi dalle approvazioni umane agli strumenti. È un meccanismo osservabile, non una garanzia che l'automazione basti a garantire la qualità.

Meta utilizza un Diff Risk Score per stimare la probabilità che una modifica provochi un incidente e alimentare procedure calibrate sul rischio. [8]

Cloudflare trasforma standard espliciti di Ingegneria in verifiche automatiche. Alcune regole obbligatorie possono impedire l'approvazione di una modifica finché non venga corretta o sia concessa un'eccezione esplicita. [9]

È un cambiamento importante perché modifica il ruolo degli esperti.

Un Garante tecnico non deve ricordare manualmente la stessa regola in cinquanta pull request se è possibile verificarla automaticamente.

Un Garante del Design non deve approvare ogni pulsante se il Design System rende facile la scelta corretta e visibile quella sbagliata.

Un Garante del Prodotto non deve rispiegare ogni settimana gli stessi vincoli se le decisioni strutturali, i clienti obiettivo e gli scopi sono accessibili.

Le regole diventano infrastruttura.

Test, CI, Design System, standard, regole di sicurezza, modelli, istruzioni per agenti, metriche e contesto condiviso si fanno carico di una parte dei controlli che prima spettavano alle persone.

Gli esperti possono così concentrarsi sulle eccezioni.

Non tutte le decisioni meritano lo stesso controllo

È qui che il rischio diventa centrale.

Una traduzione e una modifica al sistema dei permessi non dovrebbero seguire la stessa procedura di validazione.

Un cambiamento locale e reversibile non richiede lo stesso livello di controllo di una migrazione di dati o di una decisione che cambia il posizionamento dell'azienda.

Un modo per comprendere queste organizzazioni è sostituire la domanda binaria — «serve un'approvazione?» — con una graduale: **quale livello di controllo giustifica questo rischio?**

I tre livelli seguenti sono **una proposta di lettura di questo libro**, non una pratica documentata in tutte le aziende citate.

Autonomia locale

La decisione rimane entro obiettivi e standard conosciuti.

È reversibile, le possibili conseguenze sono limitate e non modifica una regola strutturale.

L'owner decide.

Validazione mirata

La decisione introduce una novità, un'eccezione o un rischio significativo in un ambito.

Interviene il Garante interessato: Sviluppo, Design, Prodotto o altro.

Non si assume necessariamente la gestione del progetto. Apporta il giudizio specialistico che manca.

Coinvolgimento di un livello superiore

La decisione modifica i limiti stessi del sistema.

Sul piano tecnico: architettura strutturale, sicurezza, dati sensibili, migrazioni critiche, rischi normativi.

Per il Design: una nuova interazione importante, una rottura con il Design System o conseguenze rilevanti su altri percorsi.

Per il Prodotto: cambiamenti del cliente obiettivo, della proposta di valore, dei prezzi, di investimenti importanti, della roadmap o delle decisioni strategiche.

In questo caso l'owner non sta più decidendo semplicemente **all'interno** delle regole.

Sta proponendo di cambiarle.

Nel modello proposto qui, è il genere di situazione che richiede l'intervento di un Garante o di un livello decisionale superiore.

Senza limiti espliciti emergono due rischi opposti

L'assenza di limiti decisionali chiari espone l'organizzazione a due rischi opposti.

Il primo è la burocrazia.

Tutto deve salire di livello. Gli esperti diventano colli di bottiglia. I responsabili devono approvare decisioni banali e l'autonomia scompare.

Il secondo è la diluizione delle responsabilità.

Ognuno decide localmente senza sapere quando una decisione smette di essere locale. Si accumulano scelte contraddittorie fino a creare un problema visibile per il cliente o per l'azienda.

Su Malorean ho descritto diversi strumenti per la coerenza organizzativa che servono proprio a chiarire questi confini: linguaggio condiviso, documenti di una pagina, gerarchia delle metriche, registro delle decisioni e regole per coinvolgere livelli superiori. [10]

Il loro valore non consiste nel produrre più documentazione.

Consiste nel rendere esplicite alcune domande:

- chi decide?
- su quali questioni?
- in quale momento?
- che cosa può restare a livello locale?
- che cosa deve essere portato a un livello superiore?
- in quali condizioni una decisione può essere riaperta?

Questi due rischi conducono a un'ipotesi organizzativa: **collocare meglio le decisioni** potrebbe contare più che moltiplicare o eliminare approvazioni.

Decidere significa anche accettare una rinuncia

I meccanismi di protezione non possono essere solo tecnici.

Una parte essenziale delle regole riguarda la capacità di compiere scelte vere.

Le organizzazioni spesso prendono decisioni così vaghe da mantenere l'apparenza del consenso. Ciascuno può interpretarle secondo le proprie priorità, e soltanto durante l'esecuzione si scopre che i vincoli erano incompatibili.

Le chiamo **decisioni deboli**. [11]

Sono particolarmente pericolose in un sistema molto autonomo.

Se l'obiettivo è ambiguo, ogni team può prendere una decisione localmente ragionevole ma globalmente incoerente.

Concedere più autonomia richiede quindi di rendere visibili le rinunce accettate.

Quale cliente viene prima?

Quale segmento non serviremo per ora?

Quale livello di qualità è irrinunciabile?

A quale vincolo possiamo rinunciare temporaneamente?

Quale obiettivo prevale quando entra in conflitto con un altro?

L'autonomia non ha bisogno di regole che dicano tutto.

Ha bisogno di un quadro che renda abbastanza chiare le vere scelte, affinché le decisioni locali convergano.

Misurare il risultato diventa l'ultimo meccanismo di controllo

L'autonomia non termina quando qualcosa viene consegnato.

Se una persona è davvero responsabile di un progetto, deve anche osservare ciò che accade dopo.

Il primo capitolo distingueva già la quantità prodotta dal valore creato. Ora la questione è diversa: **come fa chi ha preso una decisione a sapere se era quella giusta?**

Canva offre un meccanismo concreto: i suoi team possono collegare gli esperimenti a una stima del loro impatto commerciale. [12]

Per ogni progetto, questo porta a precisare:

- quale cambiamento ci si aspettava;
- quale misura permette di osservarlo;
- quale livello di qualità doveva essere mantenuto;

- che cosa hanno fatto davvero gli utenti;
- se sia opportuno continuare, correggere, abbandonare o approfondire.

Misurare chiude il ciclo della responsabilità.

Senza misurazione, l'autonomia rischia di diventare un semplice permesso di produrre.

Con essa, diventa responsabilità sul risultato.

L'autonomia non significa che il management si ritiri

Il modello di team autonomi di Marty Cagan, presentato nel capitolo 1.4, aiuta a chiarire il ruolo del management. Affidare un problema a un team senza consegnargli una soluzione già definita richiede obiettivi e contesto strategico chiari. [13] Cagan insiste inoltre sullo sviluppo delle competenze, sull'accompagnamento e sulla leadership: lasciare che il team decida non significa abbandonarlo. [14]

Il confronto ha però un limite importante. Cagan descrive la responsabilità dei team di Prodotto composti da competenze complementari. Il concetto di Garante che propongo in questo libro risponde a un altro problema: come preservare la responsabilità di un ambito quando l'esecuzione passa da una professione all'altra?

Il Garante custodisce le regole, non approva ogni gesto

Il capitolo 1.6 presentava il Garante come responsabile della coerenza di un ambito anche quando non svolge direttamente ogni compito. Ora il suo ruolo diventa concreto: definire regole comprensibili, automatizzare i controlli ripetitivi e intervenire quando il rischio o la novità supera ciò che un team può decidere da solo.

Se deve ispezionare tutto, torna a essere un collo di bottiglia. Se nessuno risponde più di nulla, la responsabilità si diluisce.

L'autonomia, quindi, non consiste né nell'approvare ogni azione né nell'eliminare tutti i controlli: consiste nel rendere possibili più decisioni locali mantenendo chiare le responsabilità.

Questa trasformazione solleva però una domanda che va oltre Prodotto e Ingegneria.

Se il contesto circola direttamente, se le decisioni possono essere prese più vicino al lavoro e se gli agenti assorbono parte del coordinamento, **che cosa accade ai livelli di management che prima svolgevano quelle funzioni?**

Malorean Corp. ci mostrerà prima come questa questione riguardi persone concrete. Poi ne esamineremo le conseguenze per l'intera azienda.

Bibliografia

[1] How Cursor Is Navigating the New Shape of Product Craft – Atlassian / Cursor

- URL: https://www.atlassian.com/blog/how-we-build/how-cursor-is-navigating-the-new-shape-of-product-craft
- Mostra come obiettivi, controlli e giudizio di Prodotto permettano a ruoli più permeabili di agire senza allontanarsi dalla direzione scelta.

[2] Issue tracking is dead – Linear

- URL: https://linear.app/next
- Sostiene un sistema in cui feedback dei clienti, strategia, decisioni e codice condividono più contesto rispetto a una catena centrata sui ticket.

[3] Le backlog n'est pas un dépotoir : c'est un outil d'action – Malorean

- URL: https://malorean.net/articles/2026-06-03-le-backlog-nest-pas-un-depotoir.it.html

- Distingue il backlog come strumento temporaneo di coordinamento dalla memoria duratura, dal contesto e dal ragionamento di Prodotto.

[4] 5 insights from Ramp's CPO on building for hypergrowth – Ramp

- URL: https://ramp.com/blog/5-insights-from-ramps-cpo-reddit-ama
- Sottolinea ICP e problemi prioritari e la necessità di rifiutare opportunità che disperdono le risorse del Prodotto.

[5] What It Takes to Build Great Products Now – Figma

- URL: https://www.figma.com/blog/what-it-takes-to-build-great-products-now/
- Ricorda che creare rapidamente un prototipo non basta: casi reali, eccezioni e qualità restano fondamentali per un prodotto duraturo.

[6] Le PM comme architecte du Contexte – Malorean

- URL: https://malorean.net/articles/2026-06-08-le-pm-comme-architecte-du-contexte.it.html
- Propone di trasformare segnali, fonti, decisioni e scelte sparse in un contesto riutilizzabile per migliorare le decisioni future.

[7] Product Decision Record : tracer les choix produit qui structurent l'entreprise – Malorean

- URL: https://malorean.net/articles/2026-06-03-product-decision-record-tracer-les-choix-produit.it.html
- Definisce una memoria sintetica delle decisioni strutturali: contesto, alternative, scelta, conseguenze e condizioni di revisione.

[8] Diff Risk Score: AI-driven risk-aware software development – Meta

- URL: https://engineering.fb.com/2025/08/06/developer-tools/diff-risk-score-drs-ai-risk-aware-software-development-meta/

- Presenta un punteggio di rischio che permette di adattare i controlli alla probabilità di incidenti causati da una modifica.

[9] How Cloudflare enforces engineering standards using AI – Cloudflare

- URL: https://blog.cloudflare.com/engineering-standards-enforcement/
- Mostra come standard espliciti diventino controlli automatici in grado di bloccare alcune violazioni prima della produzione.

[10] Les outils de cohérence organisationnelle – Malorean

- URL: https://malorean.net/articles/2026-06-02-les-outils-de-coherence-organisationnelle.it.html
- Collega linguaggio comune, obiettivi, metriche, memoria delle decisioni e regole di coinvolgimento per chiarire le scelte organizzative.

[11] Pourquoi les organisations préfèrent les décisions molles – Malorean

- URL: https://malorean.net/articles/2026-06-02-pourquoi-les-organisations-preferent-les-decisions-molles.it.html
- Analizza decisioni apparentemente consensuali che nascondono rinunce, diluiscono responsabilità e spostano le contraddizioni nell'esecuzione.

[12] Measuring Commercial Impact at Scale – Canva

- URL: https://www.canva.dev/blog/engineering/measuring-commerical-impact-at-scale/
- Descrive una misura autonoma dell'impatto commerciale che collega esperimenti locali, decisioni di Prodotto e creazione di valore.

[13] Team Objectives - Empowerment – Marty Cagan, Silicon Valley Product Group (2020)

- URL: https://www.svpg.com/team-objectives-empowerment/

- Sostiene l'assegnazione di problemi da risolvere ai team, accompagnati dal contesto strategico necessario alle decisioni locali.

[14] Product Leadership Is Hard – Marty Cagan, Silicon Valley Product Group (2020)

- URL: https://www.svpg.com/product-leadership-is-hard/
- Spiega perché i team di Prodotto autonomi richiedano una leadership più esigente e un accompagnamento attivo anziché un management assente.

9 – Malorean Corp. – Due decisioni ragionevoli

Il lunedì mattina Vivien, responsabile del coordinamento dei due team di sviluppo, cominciò la riunione con la stessa domanda che poneva da anni.

— Chi mi aggiorna sui progetti di questa settimana?

Aveva già aperto il quadro di monitoraggio. Ticket, responsabili, scadenze e blocchi erano ordinati nelle colonne che aveva perfezionato pazientemente. Conosceva i due team di cinque sviluppatori, i punti di forza, le abitudini, i progetti su cui bisognava insistere e quelli che era meglio lasciar maturare.

Nessuno rispose subito.

Poi prese la parola Lucas.

— Per la tracciabilità ho aggiornato il Mini PRD venerdì. Abbiamo provato con il cliente la ricerca per lotto. Il prossimo passo è verificare i casi in cui i dati sono incompleti. Ho annotato le questioni aperte nel contesto.

— E la data di consegna? — chiese Vivien.

— Dipende ancora dai controlli sulle giacenze storiche. Thomas deve guardare un punto insieme a me.

Thomas annuì. Da parte sua, lavorava a un miglioramento del processo di preparazione delle spedizioni. Aveva aggiornato anche lui l'avanzamento nello strumento condiviso.

Vivien passò agli altri sviluppatori. Le risposte si somigliavano. Ognuno conosceva meglio il proprio problema, aveva preso alcune decisioni e sapeva spiegare il passo successivo. Qualcuno chiedeva ancora aiuto per

suddividere il lavoro. Gli altri non avevano aspettato la riunione per andare avanti.

Alla fine Vivien guardò il suo quadro.

Tutto ciò che aveva appena sentito era già scritto lì, a volte con più dettagli.

— Bene — disse. — Abbiamo finito.

La riunione era durata dodici minuti.

Prima gliene sarebbero serviti quarantacinque.

Vivien non sapeva bene se esserne contento.

Lavorava a Malorean da abbastanza tempo da aver conosciuto tutti gli sviluppatori insieme nell'ufficio di Le Havre. Aveva organizzato le consegne, gestito i ritardi, negoziato le priorità con il Prodotto e spiegato al direttore perché una richiesta promessa per venerdì sarebbe stata pronta solo il martedì dopo.

Quando l'azienda era passata al lavoro da remoto dopo la pandemia, aveva adattato il suo metodo. Non bastavano più le conversazioni in corridoio. Aveva chiesto di mettere per iscritto le decisioni, rendere visibili i blocchi e registrare gli accordi con gli altri reparti.

Era diventato il suo lavoro quotidiano: far circolare informazioni che nessuno vedeva nella loro interezza.

Gli piaceva. Amava soprattutto i momenti in cui un progetto mal avviato ritrovava la strada grazie a qualcuno che aveva messo in contatto le persone giuste.

Da quando i team avevano cominciato a usare gli strumenti di IA, però, qualcosa era cambiato. Gli sviluppatori producevano più in fretta, ma soprattutto alcuni avevano iniziato ad assumersi i progetti in modo diverso.

Lucas parlava direttamente con i clienti. Alice condivideva un contesto di Prodotto che spiegava le ragioni delle decisioni, non soltanto l'elenco delle funzionalità da sviluppare. Bob cominciava ad adottare lo stesso metodo.

E ora un agente preparava per Vivien parte dei riepiloghi che un tempo costruiva a mano.

Il lavoro non era scomparso. Gli sfuggiva a pezzi o veniva svolto altrove.

Non ne aveva ancora parlato con nessuno.

Il mercoledì successivo Lucas mostrò ad Alice una nuova versione della dimostrazione di tracciabilità.

— Abbiamo provato che cosa succede quando un fornitore segnala un lotto potenzialmente difettoso. Riusciamo a identificare le produzioni di cui abbiamo dati. Ho anche aggiunto uno stato «in verifica» per evitare di far credere che non ci sia nulla da controllare.

Alice osservò lo schermo.

Un avviso indicava che la situazione doveva essere esaminata. Per le produzioni più vecchie, un secondo messaggio spiegava che alcune informazioni non erano disponibili.

— È più onesto della prima versione — disse.

— Sì. Il responsabile della qualità del cliente ci ha chiesto proprio di non confondere «nessun problema identificato» con «non abbiamo le informazioni necessarie».

Alice approvò. Era una distinzione di cui avevano già parlato nel documento di contesto.

— L'hai messo nel Mini PRD?

— Sì. E ho aggiunto dei casi di test.

— Allora continua.

Alice non dovette decidere dove collocare l'avviso né il testo esatto del messaggio. Lucas sapeva quale risultato cercavano. Aveva scelto una presentazione e previsto di provarla con il cliente.

Poco dopo, Thomas presentò a Vivien un miglioramento del modulo spedizioni.

Il problema era diverso. Presso alcuni clienti un ordine poteva rimanere fermo a causa di stati intermedi o compilati male, anche quando non c'era un divieto esplicito. I team logistici perdevano tempo a far correggere pratiche in cui non era stato identificato alcun rischio.

Thomas aveva quindi chiarito le regole di preparazione: un lotto contrassegnato come «bloccato» impediva la spedizione; le altre situazioni restavano visibili e seguivano il percorso abituale, con i controlli già esistenti.

— Sei sicuro di non cambiare il comportamento degli altri moduli? — chiese Vivien.

— Per i casi già esistenti, sì. Ho aggiunto test di regressione. Non ho toccato le regole di blocco.

Vivien non aveva motivo di dubitare della prudenza di Thomas. A volte conosceva eccezioni che persino la vecchia documentazione aveva dimenticato.

— Molto bene. Fai revisionare la modifica prima di procedere.

Thomas annuì.

Le due modifiche non erano ancora state provate insieme nello stesso scenario.

Il problema emerse venerdì, in un ambiente di test integrato.

Una persona dell'assistenza aveva preparato un percorso completo: ricezione di una materia prima, produzione, segnalazione del fornitore, identificazione dei prodotti interessati e preparazione di una spedizione.

Lucas seguiva la dimostrazione in videoconferenza. C'era anche Thomas. Alice aveva chiesto di essere invitata per osservare come il futuro funzionamento sarebbe stato compreso da qualcuno che non lo aveva progettato.

Per i primi minuti andò tutto bene.

Il numero del lotto venne trovato. Comparvero le produzioni. Si mostrò l'avviso «in verifica».

Poi la persona dell'assistenza passò al modulo spedizioni.

Il documento di consegna rimaneva in preparazione. Nessun messaggio impediva di proseguire.

— Un momento — disse Alice. — Stiamo parlando dello stesso lotto?

Lucas si avvicinò allo schermo.

— Sì. Perché?

— Qui sopra è «in verifica», ma in quest'altro modulo potremmo procedere come se non ci fosse nulla di particolare.

Thomas aprì il dettaglio dell'ordine.

— Non è bloccato. È in verifica. Sono due stati diversi.

— Per me — rispose Lucas — «in verifica» significava proprio che non dovevamo considerarlo approvato finché qualcuno non avesse deciso.

Thomas si girò verso la videocamera.

— Allora perché non l'hai segnato come «bloccato»?

— Perché non volevo inventare un divieto generale. Alcuni clienti devono poter indagare senza fermare tutta l'attività. Spetta al loro responsabile della qualità decidere quali prodotti siano davvero interessati.

— E nel mio modulo — ribatté Thomas — abbiamo lotti vecchi con dati incompleti. Se comincio a bloccare tutto ciò che non è stato esplicitamente approvato, alcuni ordini non partiranno mai.

Alice guardò i due schermi.

Capiva le ragioni di Lucas. Capiva anche quelle di Thomas.

Il problema era che un utente poteva leggere «in verifica» in una parte dell'ERP e non incontrare alcun avviso che fermasse il processo in un'altra.

— Interrompiamo qui la dimostrazione — disse.

Nessuno protestò. Non erano stati toccati dati reali dei clienti e le modifiche non erano arrivate in produzione. Era proprio a questo che serviva il test.

Ma il silenzio che seguì era meno confortevole di quello dopo una dimostrazione riuscita.

Vivien si unì alla chiamata quando gli spiegarono ciò che era accaduto.

Lasciò che Lucas descrivesse la propria decisione e Thomas raccontasse la sua. Fece qualche domanda sui test e sul perimetro di ogni modifica.

— Avete entrambi il vostro Mini PRD? — chiese.

— Sì — rispose Lucas.

— E le regole per decidere?

— Sì — disse Thomas. — Non ho cambiato il blocco esistente. Era un miglioramento locale del percorso di preparazione.

— Nemmeno io — aggiunse Lucas. — Ho aggiunto informazioni per non nascondere un dubbio. Avevamo proprio scritto che era importante.

Vivien aprì i due documenti fianco a fianco.

Non erano sbagliati. Semplicemente parlavano di cose diverse.

Il primo descriveva come segnalare un'incertezza nella tracciabilità. Il secondo stabiliva quando una spedizione dovesse fermarsi. Nessuno dei due rispondeva alla domanda che li collegava: che cosa doveva fare il software quando un lotto era sospetto, ma non ancora esplicitamente bloccato?

— C'è un vuoto tra i due progetti — disse Vivien.

Thomas annuì.

— Tecnicamente possiamo scegliere diversi comportamenti. Ma dobbiamo sapere quale vogliamo.

— E dobbiamo chiedere al cliente che cosa fa davvero il suo responsabile della qualità in quel caso — aggiunse Lucas. — Ho delle ipotesi, non una regola valida per tutti.

Alice intervenne.

— Chiarirò quale comportamento ci aspettiamo dal prodotto. Ma non da sola. Ci servono assistenza, Lucas, Thomas e probabilmente un cliente per valutare le conseguenze.

Vivien alzò gli occhi.

— Allora vi organizzo una riunione domani.

Alice esitò.

— Puoi aiutarci a mettere insieme le persone giuste, certo. Ma non spetta a te prendere la decisione al nostro posto.

Non lo diceva per escluderlo. Eppure Vivien percepì quelle parole come uno spostamento.

Per anni era stato la persona che riportava i problemi nel circuito corretto. Questa volta, gli interessati sapevano già di doversi parlare. Non avevano bisogno che trasformasse il disaccordo in una sequenza di ticket.

Avevano bisogno di qualcos'altro: una decisione che nessuno dei due sviluppatori poteva prendere da solo.

Il giorno dopo Alice cominciò chiedendo a ciascuno che cosa volesse evitare.

Lucas non voleva che un responsabile della qualità credesse di avere una tracciabilità completa quando mancavano dati.

Thomas non voleva che una regola imprecisa bloccasse inutilmente le operazioni dei clienti con dati storici incompleti.

L'assistenza ricordò che una segnalazione di un fornitore non comportava sempre un richiamo immediato. Alcuni clienti dovevano prima identificare l'ambito del problema, consultare i loro team della qualità e decidere le azioni da intraprendere.

— Bene — disse Alice. — Forse il nostro errore è stato cercare di dare due significati diversi allo stesso stato.

Condivise una proposta.

Un lotto poteva essere indicato come **da esaminare** senza che l'ERP pretendesse di prendere al posto dell'azienda cliente una decisione normativa o relativa alla qualità. Tuttavia, i percorsi che potevano usare o spedire quel lotto dovevano rendere visibile l'allerta e richiedere la procedura di conferma prevista, invece di lasciarla sparire.

I dati storici incerti dovevano continuare a essere segnalati come tali. E i casi in cui il software non disponeva delle informazioni necessarie non potevano essere presentati come assenza di rischio.

— Questo possiamo verificarlo nel modulo spedizioni — disse Thomas. — Ma voglio provare anche gli ordini vecchi.

— E io voglio rivedere i termini con il cliente — aggiunse Lucas. — «Da esaminare» potrebbe non significare la stessa cosa per il suo responsabile della qualità.

Alice annuì.

Non avevano finito di progettare la soluzione. Avevano finalmente individuato chi doveva decidere che cosa.

Annotò la decisione provvisoria e le domande aperte nel contesto condiviso, con le ragioni e i casi da verificare. Aggiunse una condizione alla loro tabella: quando una decisione locale cambiava il significato di un'informazione usata da più moduli, non era più automaticamente verde, anche se la modifica al codice sembrava reversibile.

— Avremmo dovuto accorgercene prima — disse Lucas.

— Sì — rispose Alice. — Ma è proprio per questo che facciamo i test. Ora sappiamo che cosa guardare la prossima volta.

Thomas esaminò i due documenti.

— Ci servirà anche un test che attraversi più moduli. Ne abbiamo molti che verificano ogni comportamento separatamente. Molti meno che controllano che cosa accade quando si incontrano.

Vivien, che ascoltava da qualche minuto, prese nota.

Per la prima volta dall'inizio della chiamata ritrovava un problema che conosceva bene: nessuno aveva una visione completa di tutte le dipendenze.

Ma fu colpito dal modo in cui si stava costruendo la soluzione. Alice non aveva imposto le schermate. Lucas e Thomas non aspettavano che qualcuno dividesse le attività. L'assistenza portava l'esperienza del cliente. Il suo ruolo era stato mettere in luce la lacuna e facilitare la conversazione.

Era stato utile.

Ma non aveva quasi nulla a che vedere con le riunioni sulle consegne che organizzava ogni lunedì.

Nel pomeriggio Thomas ricevette un messaggio da uno sviluppatore meno esperto dell'altro team.

Non capiva perché una vecchia procedura di magazzino facesse due controlli apparentemente identici. L'agente di programmazione proponeva di eliminarne uno.

Thomas aprì il file.

— Non toccare niente finché non l'abbiamo guardato insieme — rispose.

Passarono mezz'ora condividendo lo schermo. Il secondo controllo copriva un caso ereditato da una vecchia evoluzione del prodotto, spiegato male nel codice. Thomas non ricordava esattamente a quale rilascio risalisse, ma sapeva dove cercare.

— Il tuo strumento non si è inventato una risposta sbagliata — spiegò. — Ha visto due controlli simili e ha proposto di semplificare. Senza conoscere la storia del programma capisco perché.

— Ma tu non usi mai l'IA per programmare? — chiese lo sviluppatore.

— Non per farmi proporre modifiche alla cieca su questioni come questa.

— Però hai appena passato trenta minuti su una cosa che forse Lucas avrebbe verificato in cinque.

Thomas rifletté.

— Può darsi. E forse Lucas avrebbe trovato la stessa cosa. Ma conta che tu capisca perché esiste il controllo. La prossima volta saprai porre una domanda migliore, sia allo strumento sia a me.

Lo sviluppatore chiese se poteva documentare la spiegazione.

— Assolutamente sì. Sono stufo di essere l'unico a ricordarmelo.

Thomas chiuse la chiamata, sorpreso dalla propria frase.

Per anni la conoscenza delle eccezioni lo aveva reso indispensabile. A volte aveva temuto che l'IA rendesse meno preziosa quella esperienza. La conversazione gli aveva mostrato un'altra possibilità: poteva aiutare gli altri ad acquisire il giudizio che mancava loro, anziché tenere ogni risposta per sé.

Aprì il documento condiviso e scrisse qualche riga.

Durante l'incontro del lunedì Hubert chiese ad Alice come procedesse l'esperimento.

Lei raccontò le due decisioni incompatibili, il test che aveva rivelato il problema e la correzione delle regole.

— Quindi avete lasciato decidere due sviluppatori e per poco non hanno creato un'incoerenza? — riassunse Hubert.

— Sì.

— Non è molto rassicurante.

— L'abbiamo scoperta prima di mettere qualcosa in produzione — rispose Alice. — Il problema non è che due persone abbiano preso decisioni. È che nessuna delle nostre regole spiegava come dovessero convivere.

Hubert guardò gli appunti sull'incidente.

— Prima probabilmente avremmo organizzato un'altra riunione di approvazione.

— Forse. Ma una riunione in più non avrebbe necessariamente fatto emergere il problema. Serviva qualcuno che provasse l'intero percorso e responsabilità chiare quando una decisione coinvolge più ambiti.

— E adesso chi decide?

— Lucas porta avanti il suo progetto. Thomas segue il proprio e ci aiuta sui rischi tecnici. Io sono la garante della coerenza del Prodotto. Nei casi delicati consultiamo gli specialisti competenti. Non possiamo delegare tutto questo a uno strumento.

Hubert annuì lentamente.

— E Vivien?

Alice impiegò qualche secondo a rispondere.

— Ci ha aiutato ad avvicinare i due team e capire dove si trovava il blocco. Ma non spettava a lui distribuire le attività né decidere il comportamento del prodotto.

Hubert sembrò pensare ad altro.

— L'ho visto venerdì — disse. — Mi ha mostrato il nuovo monitoraggio dei progetti. Il rapporto è ottimo. Ma credo che una macchina potrebbe preparare l'ottanta per cento di ciò che mi ha raccontato.

— Lo sa anche lui — rispose Alice.

— Non so che cosa dobbiamo concluderne.

Alice non aveva una risposta pronta.

— Non che sia diventato inutile — disse infine. — Abbiamo appena visto che sa far lavorare insieme le persone. Solo che non deve più essere necessariamente il passaggio obbligato di ogni progetto.

Hubert continuò a fissare lo schermo.

— Dovremo parlarne con lui.

Vivien ricevette l'invito per il giorno seguente.

Il primo pensiero andò a una riorganizzazione. Da qualche settimana la sua agenda aveva più spazi liberi. Gli sviluppatori seguivano i propri progetti. Gli strumenti raccoglievano automaticamente una parte delle informazioni che lui passava le serate a riunire. Continuava a gestire difficoltà umane, aiutare i nuovi arrivati e sbloccare situazioni complesse, ma quelle attività non riempivano più le giornate come prima.

Gli era piaciuto essere indispensabile per le consegne di Malorean.

Non era certo che gli piacesse scoprire di esserlo meno.

Eppure la riunione cominciò con una domanda che lo sorprese.

— Che cosa vorresti continuare a fare, Vivien? — chiese Hubert.

— Il mio lavoro — rispose.

Hubert sorrise.

— È proprio questo che stiamo cercando di capire.

Vivien guardò Alice, invitata a parlare del funzionamento dei team, poi tornò a Hubert.

— Non voglio che facciamo finta di niente. I team non hanno più bisogno che ogni settimana chieda a che punto sono. E non voglio passare il tempo a rileggere riepiloghi già preparati da un agente.

— Di che cosa hanno ancora bisogno? — chiese Alice.

— Dei conflitti che non vengono scritti da nessuna parte. Delle persone che non osano dire di essere in difficoltà. Dei nuovi che non sanno a chi rivolgersi. Delle decisioni che attraversano più team. E a volte di qualcuno che si prenda semplicemente il tempo di ascoltare.

Fece una pausa.

— Ma questo non significa che tutto debba continuare a passare da me.

Hubert si raddrizzò.

— No. E dovremo dire chiaramente chi ne risponderà se il tuo ruolo cambia. Non voglio che eliminiamo un titolo e poi scopriamo che nessuno si occupa più delle persone.

Vivien annuì.

Parlarono dell'accompagnamento degli sviluppatori, delle decisioni trasversali e di ciò che doveva rimanere sotto la supervisione della direzione. Alice raccontò come Thomas aiutasse i colleghi a comprendere meglio il vecchio codice.

Non emerse alcuna soluzione definitiva.

Finita la chiamata, Vivien rimase qualche istante a guardare il calendario.

Si sentiva meno minacciato del giorno precedente, ma nemmeno del tutto rassicurato. La sua professione non era scomparsa. Si stava dividendo in diverse responsabilità, alcune delle quali potevano essere svolte diversamente.

Quella sera spense il computer prima del solito.

Per una volta non aveva alcun ritardo nelle consegne da recuperare.

E ancora non sapeva che cosa avrebbe fatto del tempo riconquistato.

10 – L'IA farà scomparire i livelli intermedi delle aziende?

Oggi un team può consultare direttamente le decisioni che lo riguardano, agire entro regole conosciute e affidare parte del monitoraggio agli agenti. Ha meno spesso bisogno di un responsabile che trasmetta le informazioni o organizzi ogni fase del lavoro.

Allora **che cosa accade ai livelli gerarchici che svolgevano questo ruolo?**

La risposta sembra ovvia: se l'IA si occupa del coordinamento, i manager intermedi perdono la loro utilità. Eppure, un manager non si limita a trasmettere istruzioni. Decide tra alternative, accompagna le persone, collega i team e si occupa dei problemi che non rientrano in alcuna procedura.

La vera domanda, quindi, non è tanto quanti posti scompariranno, quanto capire **dove andranno le loro funzioni e il potere di decidere**.

Una gerarchia costruita per coordinare le decisioni

Un cliente importante segnala un problema. L'Assistenza lo comunica al proprio responsabile, che contatta il Prodotto. Il Prodotto chiede una stima all'Ingegneria. Diversi responsabili riuniscono le informazioni, confrontano le urgenze e talvolta portano la decisione a un livello ancora superiore.

Questo percorso non è necessariamente assurdo. Ogni responsabile possiede una parte del contesto e può impegnare una parte dell'azienda. Diventa pesante quando questa catena è l'unico modo per comprendere un problema e decidere.

Storicamente, la gerarchia ha permesso alle imprese di crescere nonostante la frammentazione delle informazioni. I livelli intermedi traducono

gli obiettivi della direzione in priorità, coordinano i team, sorvegliano le dipendenze e risolvono i conflitti sulle risorse. Partecipano anche alla valutazione dei dipendenti, alla loro crescita e alla gestione delle difficoltà umane.

Ma non tutte queste attività hanno la stessa natura. Preparare un rapporto su un incidente è diverso dal decidere di rinunciare a un'opportunità commerciale. Seguire una scadenza non richiede lo stesso giudizio necessario a risolvere un conflitto o aiutare qualcuno a migliorare.

Su Malorean avevo già difeso una distinzione simile a proposito del Product Owner: una posizione può perdere la sua ragion d'essere come semplice intermediario senza che la funzione che doveva garantire diventi inutile. Bisogna mettere in discussione il lavoro svolto, non soltanto il titolo di chi lo compie. [1]

Se diminuiscono i livelli intermedi, chi assumerà le funzioni che svolgevano?

Il lavoro a distanza aveva già cambiato le regole del coordinamento

L'IA non ha inventato questa domanda. Le aziende che lavorano a distanza, talvolta in più paesi e fusi orari, hanno dovuto imparare a decidere senza affidarsi alle conversazioni in corridoio o alle riunioni improvvisate.

In GitLab il principio *handbook-first* consiste nel rendere accessibili per iscritto le regole e il contesto. Chiunque può proporre una modifica; i responsabili dell'ambito decidono se integrarla. Un dipendente può così trovare un'informazione senza aspettare che un manager gliela comunichi. [2]

Alan descrive una pratica diversa ma complementare. Le decisioni importanti vengono discusse in thread GitHub: una persona prende la decisione e la mette in pratica, un'altra continua a rispondere del successo del pro-

getto e le persone interessate sono consultate o informate. La discussione può essere aperta senza richiedere l'unanimità. [3]

In 37signals, l'azienda di Basecamp, una parte importante del coordinamento avviene attraverso scambi scritti, riepiloghi e informazioni conservate insieme ai progetti, anziché disperse in conversazioni effimere. [4]

Non sono tre aziende senza management. Sono tre modi di **ridurre la dipendenza da una persona che possiede le informazioni o deve ricordare continuamente ciò che è stato deciso**.

Questo precedente conta anche per gli agenti di IA. Un sistema può consultare una regola, recuperare una decisione e identificare un responsabile purché l'organizzazione abbia reso esplicite quelle informazioni. Una decisione che rimane nella testa di un manager è inaccessibile tanto ai colleghi quanto agli agenti.

Attenzione però a non idealizzare il lavoro da remoto. Uno studio su oltre 61.000 dipendenti Microsoft ha osservato, tra l'altro, che le reti di collaborazione sono diventate più separate dopo il passaggio al telelavoro. Scrivere di più non garantisce né la condivisione del contesto né la cooperazione tra i team. [5]

La vera eredità del lavoro a distanza non è la scomparsa dell'ufficio: è la possibilità di organizzare una parte del lavoro intorno a informazioni e decisioni accessibili senza intermediari permanenti. L'IA può basarsi su questo risultato. Non lo crea da sola.

L'IA comincia a svolgere una parte di questo lavoro

Un documento conserva una regola; un cruscotto mostra un ritardo. Un agente può fare di più: collegare diverse informazioni, preparare un aggiornamento, rilevare una contraddizione, proporre una soluzione o eseguire un'azione autorizzata.

Sono proprio attività che occupavano parte del tempo dei manager: consolidamento, reportistica, monitoraggio e preparazione delle decisioni. Il loro costo può diminuire. Non è però dimostrato che scompaiano.

Un'analisi esplorativa di HEC Paris, basata fra l'altro su 45 interviste, suggerisce che i compiti di gestione strutturati e ricchi di dati siano più adatti all'automazione rispetto a quelli che dipendono dalla fiducia, dalle relazioni umane e dal giudizio. Non misura quante ore di management siano state davvero eliminate né quanti posti siano diventati inutili. [6]

Un esperimento condotto in Procter & Gamble mostra inoltre che, nelle attività di innovazione di prodotto, una persona che lavora con l'IA può ottenere risultati paragonabili a quelli di un gruppo di due persone senza IA. Lo strumento aiuta in particolare a combinare prospettive professionali diverse. Ma facilitare una collaborazione occasionale non significa sostituire stabilmente un responsabile. [7]

In un altro esperimento, che ha coinvolto 7.137 dipendenti di 66 aziende, gli utenti abituali di un assistente di IA hanno risparmiato circa due ore settimanali dedicate alle email nella seconda metà dello studio. Eppure, in sei mesi, i ricercatori non hanno osservato un cambiamento significativo della composizione delle attività. Lo studio riguardava un assistente per il lavoro d'ufficio, non agenti capaci di eseguire da soli processi completi. [8]

La differenza è decisiva: **un guadagno individuale di produttività non è ancora una trasformazione organizzativa**.

Affinché un livello gerarchico diventi meno necessario, le informazioni devono arrivare davvero ai team, i diritti di decisione devono cambiare e le situazioni insolite devono trovare un luogo in cui essere risolte. Non è più soltanto una questione di strumenti, ma di organizzazione.

Tre aziende, tre traiettorie molto diverse

Le organizzazioni non reagiscono tutte nello stesso modo. Tre casi permettono di distinguere una riorganizzazione già avviata, un'ambizione costruita intorno all'IA e le difficoltà di una trasformazione accelerata.

Bayer: una gerarchia realmente ridotta

Con il suo modello *Dynamic Shared Ownership* (DSO), Bayer ha trasferito parte delle decisioni a team multidisciplinari. Nella lettera agli azionisti relativa al 2025, il massimo dirigente del gruppo dichiara di aver eliminato fino a sei livelli organizzativi e ridotto di circa due terzi le posizioni di management. Dichiara anche decisioni più rapide e tempi di sviluppo ridotti. [9]

La portata merita attenzione, anche se gli effetti sono riportati dalla stessa Bayer. La trasformazione richiede di cambiare le responsabilità quotidiane, non soltanto le caselle dell'organigramma: i team prendono più decisioni locali, i responsabili dedicano più tempo ad accompagnarli e alcune funzioni di coordinamento vengono riorganizzate.

C'è però una precisazione che cambia il modo di leggere questo caso: **DSO è cominciato prima della diffusione degli agenti di IA generativa**. Bayer dimostra che un grande gruppo può ridurre notevolmente la gerarchia; non dimostra che sia l'IA a provocarlo.

Block: far circolare il contesto senza attraversare tutta la gerarchia

Block va oltre nel ruolo che intende affidare all'IA. In un testo firmato da Jack Dorsey e dall'investitore Roelof Botha, i suoi dirigenti descrivono un'organizzazione basata su un sistema informativo condiviso.

L'ambizione è che un dipendente comprenda una situazione e possa agire senza aspettare che un responsabile gli trasmetta il contesto. Gli autori distinguono chi contribuisce direttamente, i responsabili di risultati trasversali (DRI) e i responsabili che continuano a produrre mentre accompagnano gli altri, chiamati *player-coaches*. [10]

Il manager che serviva soprattutto da intermediario perderebbe così importanza. Rimangono invece necessari lo sviluppo delle persone, la responsabilità dei risultati e le decisioni della direzione.

Si tratta di un'ambizione descritta dall'azienda e dal suo investitore, non di una valutazione indipendente che provi il successo duraturo del modello.

Meta: quando la produzione supera la capacità di controllo

Il contrappunto viene da un'inchiesta Reuters sui progetti di riorganizzazione di Meta nel 2026: team più piccoli, grande spazio all'IA e riduzione prevista o già avviata di alcuni livelli. Una seconda ondata di ristrutturazione è stata infine abbandonata.

Secondo documenti interni citati da Reuters, le modifiche ai software e alle infrastrutture interne sono aumentate molto più rapidamente delle nuove funzionalità consegnate agli utenti. Nello stesso periodo sono cresciuti anche gli incidenti tecnici e il tempo necessario per risolverli. Meta ha contestato o ridimensionato alcuni aspetti dell'articolo. [11]

Questi numeri non consentono di attribuire gli incidenti all'IA o alla riduzione dei manager: più trasformazioni erano in corso contemporaneamente. Ma sollevano una domanda decisiva: a che cosa serve produrre di più se l'azienda non riesce più a verificare, stabilizzare e assorbire ciò che produce?

Bayer mostra che si può appiattire la gerarchia, Block ne propone una versione concepita per l'IA e Meta ricorda i rischi di una riorganizzazione in cui la produzione cresce più in fretta del coordinamento. Nessuna di queste traiettorie, da sola, annuncia il futuro del management.

Le funzioni del management non scompaiono con chi le svolge

Facciamo un esempio: un cliente segnala un malfunzionamento software. Finché il problema è conosciuto e le regole per risolverlo sono chiare, il

team può agire senza aspettare una successione di autorizzazioni. Con dati affidabili, può indagare, applicare una correzione e informare gli altri.

Poi due diagnosi entrano in conflitto. L'agente raccomanda un'azione e uno sviluppatore ne propone un'altra. La correzione potrebbe danneggiare un servizio vicino.

A quel punto qualcuno deve far emergere il disaccordo, riunire le competenze necessarie e decidere quale rischio accettare. L'autonomia non ha eliminato il coordinamento; ha cambiato il modo di esercitarlo.

Un esperimento di Martin Krzywdzinski e dei suoi coautori offre un indizio interessante. In una linea di produzione simulata, 87 coppie hanno affrontato guasti con l'aiuto di un sistema di IA che talvolta sbagliava. I team auto-organizzati comunicavano di più e risolvevano meglio i problemi rispetto a quelli con un capo designato. L'analisi sottolinea il ruolo decisivo della comunicazione. [12]

Due persone in un laboratorio non rappresentano una grande azienda. Ma lo studio ricorda una distinzione preziosa: **un team può lavorare senza un capo formale; non può fare a meno delle attività di coordinamento**.

Queste attività possono essere svolte dai colleghi, da un esperto, da una persona incaricata di decidere o da un responsabile gerarchico. La scelta dipende dalle competenze, dai rischi e dai disaccordi.

Resta anche tutto ciò che nessun rapporto automatico può fare da solo: accogliere una persona, aiutarla a crescere, risolvere una tensione, accompagnare una decisione difficile o riconoscere il lavoro svolto. Non tutti i manager svolgono bene queste attività, che possono anche essere distribuite diversamente. Ma qualcuno deve continuare ad averne il mandato e il tempo.

Gallup osserva che l'uso dell'IA non elimina le preoccupazioni per il lavoro e che il sostegno percepito in azienda è associato a minore preoccupazione. Questo non dimostra che occorra mantenere il manager tradizionale, ma ricorda che un'organizzazione è anche una comunità umana. [13]

Infine, chi formerà le persone che domani dovranno prendere decisioni difficili?

Su Malorean ho evocato un rischio per gli sviluppatori più giovani: quando un'IA scrive subito il codice e corregge gli errori, può eliminare occasioni di imparare a fare debug, comprendere un fallimento e sviluppare il proprio giudizio. È una preoccupazione da studiare, non la prova di una perdita generalizzata di competenze. [14]

Lo stesso problema riguarda i futuri responsabili. Si impara anche assistendo a una decisione difficile, ricevendo feedback e osservando come un team supera un conflitto. Un'azienda che elimina livelli deve prevedere altri modi di trasmettere questa esperienza.

In definitiva, ogni funzione può avere diverse destinazioni: essere automatizzata, assunta dal team, trasferita a uno specialista o portata alla direzione. **Un'attività può scomparire, cambiare mano o aggiungersi al lavoro di qualcun altro.**

Per valutare l'efficacia di un'organizzazione più piatta non basta contare i posti eliminati. Bisogna confrontare il lavoro di monitoraggio evitato con i nuovi oneri di supervisione degli agenti, risoluzione degli incidenti, accompagnamento e formazione. Non disponiamo ancora di un bilancio netto solido di questa ridistribuzione.

Un'organizzazione più piatta è davvero più autonoma?

Resta un'ultima domanda, meno visibile negli organigrammi: **chi decide quando scompaiono gli intermediari?**

Possiamo immaginare team meglio informati, liberi di risolvere più problemi senza portare ogni scelta al livello superiore. Ma possiamo anche immaginare l'opposto: meno capi sul campo e più decisioni controllate direttamente dalla direzione attraverso i sistemi.

Questa tensione esisteva già prima dell'IA generativa. In una ricerca su imprese industriali, Nicholas Bloom e i suoi coautori mostrano che gli strumenti che forniscono più informazioni ai dipendenti sono associati a una maggiore autonomia locale. Quelli che facilitano la comunicazione con i superiori, al contrario, possono favorire la centralizzazione: diventa più semplice chiedere istruzioni ai vertici. [15]

Questo precedente non dimostra nulla sugli agenti attuali. Illustra però un meccanismo: **diffondere i dati non equivale a distribuire il potere**.

I sistemi di gestione algoritmica rendono il rischio più concreto. Possono assegnare compiti, proporre priorità, misurare le prestazioni e attivare avvisi. La ricerca mostra che questi strumenti possono svolgere alcune funzioni di direzione o valutazione, anche quando manager umani continuano a essere presenti per applicarle o gestire le eccezioni. [16]

Un sistema non appare come superiore nell'organigramma. Eppure, se decide quali compiti riceve ciascuno, in base a quali criteri si valutano i risultati e quando un'azione è autorizzata, struttura una parte dell'autorità.

Dobbiamo quindi guardare le decisioni dietro gli schermi: chi fissa gli obiettivi degli agenti? Chi sceglie le regole e può cambiarle? Chi può contestarle? E quando un algoritmo sbaglia, chi ha davvero il potere di annullare ciò che ha provocato?

Un team può avere un'enorme capacità esecutiva ed essere incapace di cambiare le proprie priorità. Un altro può mantenere un manager e decidere davvero i mezzi a disposizione e le scelte locali.

Su Malorean ho già descritto quelle «decisioni deboli» che creano un'apparenza di accordo senza chiarire la scelta, la rinuncia accettata o chi ha il potere di decidere. Eliminare un livello non cambia nulla se il potere rimane altrettanto difficile da individuare. [17]

Possiamo quindi immaginare un paradosso: **meno gerarchia visibile, ma più controllo centralizzato negli strumenti**.

Non è inevitabile. Gli stessi sistemi possono aiutare i team a decidere più localmente e rendere più trasparenti le regole che distribuiscono il potere. Tutto dipende da ciò che l'azienda sceglie di delegare davvero, non dal numero di caselle eliminate dall'organigramma.

Che cosa potrebbe realmente scomparire

L'IA farà scomparire i livelli intermedi delle aziende?

Alcuni livelli potrebbero diventare meno necessari. Ma la loro scomparsa non sarebbe automatica né significherebbe la fine del management.

Le funzioni che consistono soprattutto nel trasmettere informazioni, consolidare aggiornamenti o organizzare approvazioni ripetitive possono perdere importanza. Altre, basate sulle scelte difficili, sui conflitti, sulla trasmissione dell'esperienza e sulla responsabilità, dovranno essere svolte diversamente.

Bayer mostra che è possibile ridurre drasticamente i livelli senza che l'IA ne sia una causa accertata. Block propone di affidare più coordinamento a un sistema condiviso. Meta ricorda che accelerare l'esecuzione senza misurare l'affidabilità e il carico umano può indebolire l'organizzazione. Nulla prova ancora una scomparsa generale del management dovuta agli agenti.

Il cambiamento potrebbe comunque essere profondo. Un responsabile sarebbe meno spesso un passaggio obbligato per ottenere un'informazione o un'approvazione. Potrebbe dedicare più attenzione alla coerenza, alle situazioni eccezionali e allo sviluppo delle persone. Altrove, alcune funzioni passerebbero direttamente ai team; altre sarebbero integrate nel software o concentrate nelle mani di pochi decisori.

Non è necessariamente il management a diventare inutile. È la gerarchia come meccanismo obbligato di coordinamento a essere messa in discussione.

Per uno sviluppatore la conseguenza è concreta. Quando il lavoro passa per meno intermediari, non ottiene soltanto più mezzi per produrre. Entra in contatto più diretto con le esigenze, i compromessi e i risultati di cui qualcuno dovrà rispondere.

È il cambiamento descritto lungo tutto questo primo libro: aumenta la capacità di agire, ma con essa anche la necessità di comprendere, scegliere e assumersi le proprie responsabilità.

Questo spostamento non garantisce decisioni migliori né organizzazioni più umane. Ma ci obbliga a pensarle diversamente: condividere il contesto, rendere identificabile il potere, preservare la trasmissione dell'esperienza e accettare di rispondere dei risultati.

La questione non è soltanto che cosa l'IA ci permetta di produrre oggi, ma quale azienda vogliamo costruire intorno a questa nuova capacità.

Nell'ultimo episodio torneremo a Malorean Corp. per vedere come dirigenti e team vivono, al di là dell'organigramma, le scelte che hanno fatto.

Bibliografia

[1] Le PO n'est pas un métier, c'est une fonction – et l'IA n'y change rien, elle accélère – Malorean

- URL: https://malorean.net/articles/2026-08-03-le-po-nest-pas-un-metier-cest-une-fonction.it.html
- Saggio dell'autore che distingue il lavoro utile di una funzione da un posto che serve soprattutto come intermediario tra decisione ed esecuzione.

[2] Handbook First – GitLab

- URL: https://handbook.gitlab.com/handbook/company/culture/all-remote/handbook-first/
- Presenta la documentazione come riferimento condiviso e le modifiche discusse pubblicamente, con responsabili umani delle decisioni.

[3] Comment prendre des décisions dans un environnement de travail asynchrone – Alan

- URL: https://alan.com/fr-fr/blog/healthy-business/a/decisions-asynchrones-methode-alan
- Spiega come le decisioni vengano discusse per iscritto, chi decida, chi continui a rispondere e chi debba essere consultato o informato.

[4] The 37signals Guide to Internal Communication – 37signals

- URL: https://37signals.com/how-we-communicate
- Descrive una comunicazione interna prevalentemente asincrona, con scambi scritti e contesto conservato nel tempo.

[5] The effects of remote work on collaboration among information workers – Microsoft Research

- URL: https://www.microsoft.com/en-us/research/publication/the-effects-of-remote-work-on-collaboration-among-information-workers/

- Esamina le reti di collaborazione dopo il passaggio al lavoro a distanza e segnala un maggior rischio di isolamento tra gruppi.

[6] AI Automates Routine Management and Raises the Value of Leadership – HEC Paris

- URL: https://www.hec.edu/en/dare/tech-ai/ai-automates-routine-management-and-raises-value-leadership
- Analisi esplorativa dei compiti manageriali basata su interviste e valutazioni di modelli, senza misurare causalmente i posti eliminati.

[7] The Cybernetic Teammate: A Field Experiment on Generative AI Reshaping Teamwork and Expertise – NBER

- URL: https://www.nber.org/papers/w33641
- Esperimento in Procter & Gamble sul contributo dell'IA all'innovazione e all'incontro di competenze diverse.

[8] Shifting Work Patterns with Generative AI – NBER

- URL: https://www.nber.org/papers/w33795
- Esperimento randomizzato in 66 aziende: risparmi individuali di tempo senza cambiamenti significativi nell'insieme dei compiti osservati in sei mesi.

[9] Chairman's Letter – Bayer Annual Report 2025

- URL: https://reports.bayer.com/annual-report-2025/en/to-our-stockholders/chairmans-letter.html
- Bayer dichiara una forte riduzione dei livelli e dei posti di management nella trasformazione DSO, precedente agli agenti.

[10] From Hierarchy to Intelligence – Sequoia Capital / Block

- URL: https://sequoiacap.com/article/from-hierarchy-to-intelligence

- Jack Dorsey e Roelof Botha presentano una visione del coordinamento tramite un sistema condiviso, con DRI e player-coaches.

[11] Mark Zuckerberg had a bold plan to replace Meta staff with AI. Here's how it imploded – Reuters

- URL: https://www.timeslive.co.za/news/sci-tech/2026-08-26-special-report-mark-zuckerberg-had-a-bold-plan-to-replace-meta-staff-with-ai-heres-how-it-imploded/
- Inchiesta Reuters riprodotta da TimesLIVE sulle tensioni organizzative e tecniche di una trasformazione in Meta.

[12] How team organization influences the ability to solve automation failures – AI & Society

- URL: https://link.springer.com/article/10.1007/s00146-025-02761-5
- Esperimento con coppie auto-organizzate e gerarchiche che hanno affrontato guasti dell'automazione, concentrato sul ruolo della comunicazione.

[13] Using AI More Does Not Reassure Workers, Managers Do – Gallup

- URL: https://www.gallup.com/workplace/713231/ai-not-reassure-workers-managers-do.aspx
- Indagine sulle preoccupazioni dei lavoratori per l'IA, sulla sensazione di rispetto e sul sostegno ricevuto nel lavoro.

[14] L'IA ne devrait pas seulement servir à produire dix fois plus. Elle devrait aussi nous obliger à comprendre dix fois mieux ce que nous produisons – Malorean

- URL: https://malorean.net/articles/2026-07-23-lia-produire-dix-fois-plus-comprendre-dix-fois-mieux.it.html
- Riflessione dell'autore sulle occasioni di apprendimento perse quando l'IA svolge compiti e correzioni che formavano gli sviluppatori junior.

[15] The distinct effects of Information Technology and Communication Technology on firm organization – NBER

- URL: https://www.nber.org/papers/w14975
- Distingue gli effetti opposti dell'accesso alle informazioni e delle tecnologie di comunicazione sull'autonomia decisionale locale.

[16] Algorithmic Management: From Technology to Politics and Theory – Weizenbaum Journal of the Digital Society

- URL: https://ojs.weizenbaum-institut.de/index.php/wjds/article/view/4_3_9/163
- Sintesi delle funzioni di direzione e valutazione svolte dai sistemi algoritmici e delle possibilità di intervento umano.

[17] Pourquoi les organisations préfèrent les décisions molles – Malorean

- URL: https://malorean.net/articles/2026-06-02-pourquoi-les-organisations-preferent-les-decisions-molles.it.html
- Saggio sui consensi apparenti che ritardano le decisioni, nascondono le rinunce accettate e rendono difficile attribuire le responsabilità.

11 – Malorean Corp. – Il tempo ritrovato

Due mesi dopo la dimostrazione del modulo ferie, Hubert Félix Duca aprì la riunione del lunedì senza chiedere a che punto fossero gli sviluppi.

— Allora, che cosa abbiamo imparato la settimana scorsa?

Alice condivise lo schermo.

Aveva davanti il riepilogo degli ultimi riscontri dei clienti, i risultati degli esperimenti in corso e alcune decisioni ancora aperte. Durante la notte un agente aveva preparato un primo raggruppamento delle informazioni. Alice lo aveva riletto al mattino, scartato diversi accostamenti ingannevoli e aggiunto due domande che l'agente non avrebbe potuto formulare da solo.

— Sulla tracciabilità, il cliente pilota riesce a identificare molto più rapidamente le produzioni interessate quando i dati sono completi — cominciò. — Ma rimangono casi storici per i quali dobbiamo mostrare chiaramente i nostri limiti. Proveremo una nuova presentazione con il suo responsabile della qualità.

Hubert annuì.

— E con gli altri clienti?

— Troppo presto per trarre conclusioni — rispose Alice. — È proprio per questo che non abbiamo esteso il rilascio a tutti.

Bob prese la parola. Si occupava di un altro gruppo di progetti di Prodotto e aveva appena avuto diverse conversazioni con i team di prevendita e assistenza.

— Abbiamo anche tre richieste che sembrano simili — disse. — L'agente le ha raggruppate sotto un'unica idea di funzionalità. Ma dietro ci sono

almeno due problemi diversi. Vorrei che chiarissimo a quali clienti vogliamo rivolgerci prima di decidere se investire.

— Quanti sviluppatori ti servono? — chiese Hubert.

Bob sorrise.

— Per ora nessuno.

Hubert ricambiò il sorriso.

Bob guardò una notifica e aggiunse:

— A proposito, da ieri l'assistenza sta provando un nuovo sistema per classificare le richieste ricorrenti. Una persona del team lo ha preparato con un agente, usando copie anonimizzate, per verificare se le informazioni arrivano più rapidamente agli interlocutori giusti.

— Chi l'ha avviato? — chiese Hubert.

— L'assistenza. Alice ha confermato che la prova rimaneva interna e reversibile, con una verifica umana dei risultati.

— Lo sapevi?

— Conoscevo l'idea. Non i dettagli del test.

Hubert ebbe l'istinto di chiedere perché nessuno lo avesse avvertito prima di cominciare. Si trattenne. Avevano proprio concordato che le iniziative limitate e reversibili non avrebbero più richiesto il suo consenso.

— Mi direte se funziona — disse infine. — E se qualcuno dovrà mantenerlo, voglio sapere chi.

Non aveva dovuto autorizzare la prova. Rimaneva responsabile delle risorse da dedicare a un'eventuale estensione.

Non molto tempo prima, una risposta del genere sarebbe sembrata strana in una riunione di Prodotto. Si cercava soprattutto di stabilire che cosa gli sviluppatori dovessero realizzare e in quale ordine. Adesso Alice e Bob

dedicavano più tempo a scegliere i problemi da risolvere, verificare la coerenza delle iniziative e intervenire quando una decisione superava i limiti stabiliti.

I titoli dei loro incarichi non erano cambiati. Nei documenti amministrativi erano ancora Product Manager.

Ma il loro lavoro quotidiano non assomigliava più del tutto a quello di due mesi prima.

Il cambiamento più visibile era forse proprio ciò che non sorprendeva più nessuno.

Uno sviluppatore parlava con un cliente. Un altro proponeva un Mini PRD per migliorare il proprio modulo. Una commerciale preparava una variante di dimostrazione nell'ambiente destinato a questo scopo. Una persona dell'assistenza segnalava direttamente un'incoerenza in una regola di Prodotto.

Non provocava più grandi discussioni sui confini tra professioni.

Malorean non aveva annunciato che tutti potessero fare tutto. Aveva semplicemente smesso di dare per scontato che ogni attività dovesse passare obbligatoriamente dalla persona che tradizionalmente occupava quel ruolo.

I due team di cinque sviluppatori esistevano ancora. La maggior parte dei loro membri gestiva ormai alcuni progetti dall'inizio alla fine: comprendere il problema, proporre l'impostazione, parlare con le persone interessate, sviluppare, testare e poi osservare i risultati.

Alcuni avevano adottato questo modo di lavorare più rapidamente di altri. Due sviluppatori preferivano ancora un accompagnamento ravvicinato per definire i progetti. Un altro era eccellente sui problemi tecnici

complessi, ma non aveva alcuna voglia di moltiplicare gli incontri con i clienti. Nessuno cercava di trasformarli a forza in copie di Lucas.

C'erano ancora i ticket.

Si usavano per coordinare un intervento, ricordare un compito preciso o seguire una correzione. Ma non si cercava più di farci entrare il motivo di un progetto, tutte le decisioni passate e ogni informazione utile a un agente.

Al loro posto, ogni responsabile di progetto manteneva uno spazio di contesto nello strumento documentale comune. Vi si trovavano obiettivi, decisioni, fonti, rischi, domande aperte e collegamenti a quanto esisteva già: ticket, preventivi, riscontri dell'assistenza, verbali degli incontri, test e talvolta vecchie discussioni recuperate.

Non serviva ricopiare tutto. L'importante era che chiunque potesse capire perché fosse stata presa una decisione, dove controllare un'informazione e chi avesse l'autorità per cambiarla.

Il primo documento di Alice era servito da esempio.

Bob aveva adottato il metodo, poi gli sviluppatori, poi diverse persone dell'assistenza, delle vendite e dell'implementazione. Molti avevano scoperto, come Alice, di poter passare un'ora a spiegare oralmente a ChatGPT decisioni che da soli avrebbero impiegato giorni a formalizzare.

Non tutto ciò che era nella testa delle persone era diventato miracolosamente accessibile. Ma una parte delle ragioni, delle eccezioni e delle conoscenze tacite cominciava a esserlo.

Il lavoro da remoto aveva reso questa evoluzione più naturale. Dopo la pandemia, Malorean aveva già imparato a scrivere di più affinché colleghi sparsi per la Francia potessero lavorare senza aspettare una conversazione

in corridoio. L'IA aveva accelerato la raccolta e l'organizzazione delle conoscenze. Non aveva inventato il bisogno di condividerle.

Alice aprì una pagina intitolata *Decisioni di Prodotto — regole comuni*.

— La settimana scorsa abbiamo avuto un altro equivoco — spiegò a Bob. — Nella documentazione di un team, «approvato» significava «revisionato tecnicamente». Per l'assistenza significava «comportamento confermato con il cliente».

Bob sospirò.

— Quella parola non ha mai significato la stessa cosa per tutti.

— Appunto. Ho aggiunto un rimando alla decisione presa dopo il problema dei lotti.

Qualche settimana prima Lucas e Thomas avevano rischiato di far convivere due comportamenti incompatibili tra tracciabilità e spedizioni. Le modifiche erano state fermate durante i test. Da allora, i team avevano imparato a diffidare dei termini che sembravano ovvi in un modulo ma cambiavano significato in un altro.

Il contesto era stato d'aiuto. Ma aveva anche creato nuovi problemi.

Alcune pagine non erano più aggiornate. Decisioni provvisorie erano state copiate come se fossero regole definitive. Una volta un agente aveva recuperato una vecchia ipotesi commerciale presentandola come un impegno assunto dall'azienda.

Avevano dovuto stabilire una disciplina: ogni contesto indicava il proprio responsabile, le fonti, le decisioni confermate e i punti ancora incerti. I cambiamenti strutturali dovevano essere registrati; le informazioni sensibili non erano indiscriminatamente accessibili a tutti gli strumenti.

— Forse avremmo potuto evitare di moltiplicare le pagine — osservò Bob.

— Forse. Ma prima esistevano anche le contraddizioni. Solo che erano nelle nostre teste e nessuno poteva vederle.

Alice rilesse l'ultima modifica del collega.

— Approvo il principio, ma non la formulazione sui clienti tedeschi: ancora non ne sappiamo abbastanza.

Bob corresse la riga.

Il lavoro del Garante spesso assomigliava a questo. Non approvare ogni schermata, ma impedire che un'ipotesi diventasse inavvertitamente una regola.

Nell'altro team, Thomas aveva appena terminato una conversazione con uno sviluppatore arrivato da poco.

Gli aveva dedicato un'ora per spiegare il codice storico dell'ERP. Il nuovo sviluppatore aveva usato un assistente per capire un vecchio meccanismo di gestione delle paghe e poi aveva chiesto perché un'eccezione sembrasse contraddire la regola generale.

Thomas aveva ritrovato il motivo di quell'eccezione in una vecchia consegna.

— Se ti fidi solo del riepilogo, rischi di pensare che il codice vada ripulito — disse. — Ma prima di ripulirlo bisogna capire che cosa stiamo per togliere.

— Potresti mettere questa spiegazione nel contesto del team?

— È quello che sto facendo.

Thomas condivise la nuova nota. Aveva aggiunto la fonte, la spiegazione e due casi di test.

Non affidava ancora la scrittura delle sue modifiche tecniche agli agenti, come faceva Lucas. Il suo scetticismo non era svanito. Eppure usava volentieri i nuovi spazi di documentazione e collaborava di più con gli altri per trasmettere la propria esperienza.

Vivien, che aveva seguito l'inserimento del nuovo sviluppatore, chiamò Thomas in videoconferenza poco dopo.

— Come va?

— Bene. È curioso. Fa domande prima di cancellare le cose.

— Rassicurante.

— E finalmente ho trovato il tempo di scrivere perché esistono certi controlli.

Vivien sorrise.

— Quindi hai cambiato idea sull'IA?

— No. Ho cambiato idea sul tenermi queste spiegazioni per me.

Fece una pausa.

— Però ci vuole tempo. Se mi chiedono di accompagnare gli altri e riempiono comunque tutte le mie giornate di sviluppo, non funzionerà.

— Ne ho parlato con Hubert. Abbiamo riservato del tempo per accompagnare gli altri nella pianificazione. Non sarà un'attività da fare solo dopo aver finito tutto il resto.

Vivien sapeva bene di cosa parlava. Per anni aveva sentito i team promettere che avrebbero formato i nuovi «appena ci fosse stato un po' di tempo». Quel momento non arrivava mai.

Parlarono ancora delle esigenze dello sviluppatore, poi Vivien chiuse la videoconferenza.

Non aveva aperto neanche una volta il quadro delle consegne.

All'ora di pranzo Hubert chiese a Vivien di collegarsi a una chiamata.

— Ho una domanda — esordì. — Quanto tempo passi oggi a fare ciò che facevi l'anno scorso?

Vivien rifletté.

— Non abbastanza da giustificare il mio vecchio posto a tempo pieno, se è questo che vuoi sapere.

Hubert non rispose subito.

Per molto tempo Vivien era stato il punto di passaggio tra due team tecnici. Gestiva le consegne, le dipendenze, una parte delle urgenze e le conversazioni tra sviluppatori e responsabili delle altre funzioni. Quando l'organizzazione era meno autonoma, quel ruolo occupava abbondantemente le sue giornate.

Non era più così.

Ogni progetto aveva un responsabile chiaramente identificato. Gli sviluppatori prendevano buona parte delle decisioni locali. Le informazioni circolavano senza che lui dovesse ritrasmetterle. Gli strumenti preparavano gli aggiornamenti e segnalavano le dipendenze, anche se bisognava ancora verificare quello che dicevano.

I problemi umani, i conflitti e l'accompagnamento non erano scomparsi. Vivien aiutava ancora a gestirli. Ma non c'era più bisogno di concentrarli in un posto dedicato alla supervisione quotidiana delle consegne.

— Non mi dispiace avere meno riunioni inutili — continuò. — Anzi. Ma non ho passato tutti questi anni a imparare a gestire i team per diventare uno che controlla se un agente ha compilato bene una tabella.

— Non te lo chiederò — rispose Hubert.

Avevano preparato una transizione. Le decisioni tecniche importanti spettavano ai Garanti competenti. Le persone incaricate di accompagnare gli altri avevano tempo riservato. Hubert manteneva la responsabilità delle decisioni di carriera e delle difficoltà umane che non potevano essere affidate a un semplice documento. Vivien avrebbe gestito il passaggio di consegne sugli ultimi progetti di coordinamento.

Non era tutto perfetto. Avevano dovuto intervenire in un caso in cui due persone credevano che l'altra stesse seguendo un collega in difficoltà. Un team aveva anche aspettato troppo una decisione perché nessuno ne aveva designato il responsabile.

Ogni volta avevano corretto le responsabilità e il modo di renderle visibili.

— C'è una cosa che non voglio — disse Hubert. — Far sparire il tuo posto e fingere che il lavoro che svolgevi non avesse alcun valore.

— Nemmeno io.

— Allora dobbiamo capire dove questa esperienza può servirci adesso.

Vivien guardò la tazza.

— Credo di avere un'idea.

Qualche giorno prima, sua moglie gli aveva parlato di nuovo della Germania.

Era tedesca. Da qualche tempo discutevano della possibilità di trasferirsi lì. Senza fretta: avevano la loro vita in Francia, le abitudini, gli amici. Ma la prospettiva tornava spesso, con dettagli sempre più concreti.

Vivien aveva sempre rimandato la decisione.

Gli piaceva troppo il lavoro in Malorean. Conosceva i team, la loro storia, il programma e i clienti. Andarsene gli avrebbe dato l'impressione di abbandonare qualcosa che aveva contribuito a costruire.

Ma amava anche le sfide. A questo punto della carriera si sorprendeva a desiderarne una nuova.

Il sabato precedente avevano guardato annunci di case in Germania, senza ammetterlo del tutto. Poi sua moglie aveva chiuso il computer.

— Sai — aveva detto — non siamo obbligati a parlarne per altri dieci anni.

Vivien aveva riso.

Ma il lunedì ci pensava ancora.

— La Germania? — ripeté Hubert.

Vivien annuì.

— Ne parliamo da anni nelle riunioni strategiche. Abbiamo alcuni clienti là, ma sono soprattutto filiali di gruppi francesi. Non abbiamo ancora imparato davvero a vendere il nostro ERP alle aziende tedesche.

— È vero.

— E continuiamo a dirci che ci andremo quando il prodotto sarà pronto. Ma non sapremo mai cosa gli manca finché non parleremo seriamente con le aziende del posto.

Hubert ascoltava con attenzione.

— Vorresti occuparti dello sviluppo di quel mercato?

— Sì. Ma senza fingere di essere diventato esperto del mercato tedesco dalla sera alla mattina. Conosco il nostro prodotto, i nostri vincoli, i nostri team e come i clienti lo implementano. So riunire le persone intorno a un problema complesso. Il resto dovrò impararlo.

— Hai mai lavorato nelle vendite?

— Non come Julie o Philippe. E non intendo spiegare loro come fare il proprio mestiere. Vorrei lavorare con loro, con chi si occupa delle implementazioni e con i nostri esperti del settore, per capire le esigenze tedesche e vedere se siamo in grado di soddisfarle.

Vivien esitò.

— Mia moglie è tedesca. Da qualche tempo parliamo di trasferirci là. Sarebbe l'occasione di fare qualcosa che mi interessa davvero senza lasciare Malorean.

Hubert rimase in silenzio per un momento.

— Sai che non si tratta semplicemente di tradurre le schermate, vero?

— Certamente. L'ERP riguarda regole e processi che cambiano da un paese all'altro. Tradurre l'interfaccia non basta. Dobbiamo capire che cosa possiamo davvero promettere e che cosa dovremo sviluppare, adattare o rifiutare.

— E potremmo sbagliarci.

— Probabilmente ci sbaglieremo su qualcosa. Preferisco scoprirlo parlando con i clienti piuttosto che dopo aver investito un anno in funzionalità che nessuno vuole.

Hubert guardò l'elenco dei progetti.

Per molto tempo avevano parlato della Germania come di un'ambizione futura. Prima dovevano concludere i progetti in ritardo, assumere più sviluppatori, liberare tempo per adattare l'ERP, preparare dimostrazioni e formare i team.

Alcune difficoltà non erano scomparse. Ma altre erano cambiate di dimensione.

Preparare una dimostrazione richiedeva meno fatica. Si potevano provare varianti del prodotto senza mobilitare sempre gli sviluppatori. Assistenza e prevendita preparavano più facilmente i documenti e condividevano le conoscenze. Agli sviluppatori mancava meno la capacità di esplorare un'idea circoscritta.

Non significava che fossero rimasti senza lavoro.

Significava che potevano scegliere di dedicare parte delle energie a qualcosa di diverso dallo smaltire le richieste dei clienti francesi.

— Preparami una proposta per una prima fase — disse Hubert. — Non una promessa di ingresso nel mercato. Degli incontri, le domande a cui vogliamo rispondere, le competenze di cui avrai bisogno e un modo per decidere se continuare.

Vivien sorrise.

— Un Mini PRD per la Germania?

— Non prendermi in giro. Se riusciamo a definire in questo modo un progetto tecnico, possiamo almeno provare a formulare le ipotesi di un progetto di mercato.

— D'accordo. Ma voglio conservare il diritto di scoprire che ci sbagliavamo.

— Ci conto.

La settimana successiva Hubert riunì Alice, Bob, Vivien, Julie e Philippe.

Presentò loro la proposta senza chiedere subito quante funzionalità avrebbero dovuto sviluppare.

— Adesso abbiamo una capacità che due mesi fa non avevamo. Vorrei usarne una parte per capire che cosa significherebbe essere davvero presenti sul mercato tedesco.

Julie fu la prima a reagire.

— Posso già preparare dimostrazioni in tedesco. Potremmo anche contattare abbastanza rapidamente alcuni potenziali clienti. Con l'IA possiamo adattare i materiali in poche ore.

— I materiali, forse — disse Philippe. — Ma un dirigente tedesco che deve acquistare un ERP non lo sceglierà solo perché la presentazione è nella sua lingua.

Julie incrociò le braccia.

— Non ho detto questo.

— Lo so. Dico soltanto che dovremo capire come scelgono i fornitori, chi consultano e di quale fiducia hanno bisogno. Ancora non conosciamo abbastanza quel mercato.

Vivien intervenne prima che la discussione diventasse un dibattito sugli strumenti.

— È proprio questo che voglio imparare. Julie, avremo bisogno della tua capacità di preparare rapidamente idee e dimostrazioni. Philippe, ci servirà la tua esperienza con i grandi clienti e con le discussioni che non si risolvono con una bella schermata. E chi si occupa delle implementazioni conosce le domande che i clienti a volte scoprono troppo tardi.

Alice intervenne.

— Da parte nostra dovremo essere molto chiari sui limiti del prodotto. Possiamo esplorare rapidamente, ma non promettere funzioni legate a norme che non abbiamo verificato.

Bob annuì.

— Dovremo anche scegliere quali aziende vogliamo conoscere per prime. «Germania» non è un tipo di cliente. Se proviamo a soddisfare tutti, ripeteremo gli errori che abbiamo imparato a evitare qui.

Hubert osservò i volti sullo schermo.

Aveva chiesto a Vivien di proporre un approccio. Nessuno gli aveva chiesto di creare subito un team permanente, riservare tutta la capacità degli sviluppatori o promettere una data di avvio commerciale.

Eppure diverse persone avevano già cominciato a capire che cosa potevano offrire.

Era diventata un'abitudine a Malorean.

— Molto bene — disse Hubert. — Questa è la mia decisione: finanziamo una prima fase esplorativa. Vivien ne sarà responsabile. Dovrà dirci quali problemi delle aziende tedesche sappiamo già risolvere, quali richiedono un adattamento e quali non vogliamo affrontare. Poi decideremo se investire di più.

Aggiunse:

— E nessuno verrà assegnato a questo progetto soltanto perché uno strumento gli ha fatto risparmiare tempo. Vedremo quanta capacità ciascuno può davvero liberare senza abbandonare i clienti che seguiamo oggi.

Vivien annotò l'ultimo punto. Aveva passato abbastanza anni a gestire calendari e piani di lavoro da sapere che la capacità liberata sulla carta non è sempre disponibile nella realtà.

Nei giorni successivi fece il giro dei team di prevendita e implementazione.

Una parte del loro lavoro era cambiata da quando le dimostrazioni si potevano preparare partendo da configurazioni riutilizzabili e i riepiloghi delle conversazioni con i clienti richiedevano meno fatica. Due persone gli raccontarono che passavano meno tempo a rifare gli stessi materiali e che avrebbero voluto partecipare alle prime interviste in Germania.

— Posso aiutarti a preparare le domande sull'implementazione — propose una di loro. — E accompagnarti a qualche incontro a distanza.

— E i tuoi clienti francesi? — chiese Vivien.

— Non posso mollare tutto. Però posso riservare del tempo se lo mettiamo in calendario.

Un'altra persona, impegnata in varie attivazioni delicate, gli disse invece di non avere alcuna disponibilità.

Vivien non insistette.

Prepararono una distribuzione provvisoria: alcuni spazi riservati per capire le aspettative delle aziende tedesche, appuntamenti con filiali già clienti quando utile, e nessuna promessa di impegno permanente prima di aver misurato lo sforzo reale.

Il tempo risparmiato non si era trasformato da solo in crescita. Bisognava ancora decidere che cosa farne e accettare di non riempire immediatamente ogni ora recuperata con le richieste di ieri.

8

Quella sera Alice ricevette un messaggio da Lucas.

Ho visto il progetto Germania. Abbiamo già un meccanismo di configurazione che potrebbe aiutarci a provare alcune differenze nei processi. Posso preparare un ambiente esplorativo senza toccare il prodotto standard. Vi interessa?

Alice rilesse il messaggio.

Due mesi prima una proposta simile avrebbe probabilmente aperto una discussione sul fatto che uno sviluppatore dovesse aspettare che gli fosse affidato un progetto. Oggi la prima domanda era più semplice: serviva alla strategia e entro quali limiti poteva procedere?

Alice aprì il contesto della prima fase tedesca. Vivien lo aveva già compilato con ipotesi, responsabili, limiti e domande aperte.

Rispose:

Sì, se definisci il progetto insieme a Vivien e se non trasformiamo un'ipotesi di business in un comportamento standard senza validazione. Parti da ciò che vogliamo imparare, non da ciò che sappiamo costruire.

Lucas rispose quasi subito.

Ricevuto. Preparo un Mini PRD.

Alice sorrise.

Ricordò il venerdì in cui aveva mostrato loro un modulo ferie sviluppato in due giorni. Allora tutti cercavano di capire come avesse potuto produrre così tanto e così in fretta.

Quella sera non sorprendeva più nessuno che potesse costruire rapidamente una dimostrazione.

La questione era capire a che cosa sarebbe servita.

Vivien concluse la giornata condividendo con sua moglie le date dei primi incontri previsti con aziende tedesche.

Non aveva ancora firmato alcun contratto. Malorean non aveva aperto un ufficio né adattato l'ERP a tutte le esigenze del mercato. Avrebbero dovuto scoprire le pratiche locali, verificare i vincoli giuridici e normativi, imparare a convincere aziende che non li conoscevano.

Le incognite erano molto più numerose delle certezze.

— Ti fa paura? — chiese sua moglie.

— Un po'.

— Sembravi così sicuro di te quando coordinavi i tuoi team.

Vivien sorrise.

— Avevo passato anni a imparare quel mestiere.

— E adesso?

Guardò il documento che aveva cominciato a preparare per la prima visita.

Domande. Ipotesi. Nomi delle persone da incontrare. Un primo elenco di cose che non sapeva.

— Adesso — rispose — ho di nuovo qualcosa da imparare.

Lei si avvicinò a guardare lo schermo.

— E noi, in tutto questo?

— Possiamo finalmente parlare sul serio del nostro trasferimento.

Gli posò una mano sulla spalla.

Vivien ripensò al vecchio quadro delle consegne, alle riunioni del lunedì, a tutto il lavoro che aveva tenuto insieme i progetti. Gli era piaciuto. Non rimpiangeva quegli anni.

Non rimpiangeva neppure di aver accettato che i team non avessero più bisogno di lui nello stesso modo.

Sullo schermo era confermato il primo appuntamento in Germania.

Per Malorean non era ancora una conquista né un successo.

Era un inizio.

12 – Conclusione

Professioni con confini meno netti, responsabilità più visibili

Le professioni non delimitano più con precisione ciò che ciascuno può fare. Uno sviluppatore può esplorare un'esigenza, un PM può costruire una soluzione, un commerciale può realizzare un prototipo. L'IA apre possibilità che prima erano riservate ad altre specializzazioni. Non rende però le loro competenze intercambiabili.

Non siamo più responsabili soltanto di ciò che eseguiamo personalmente. Possiamo seguire un progetto che attraversa più funzioni, mentre i Garanti vigilano sulla coerenza dei rispettivi ambiti. La capacità di agire si diffonde; la responsabilità non deve dissolversi.

Il rischio è duplice: continuare a dire «non è il mio mestiere» oppure presumere «se posso farlo, posso anche decidere da solo». Né difendere i confini professionali né cancellarli basta a organizzare il lavoro.

L'autonomia si fonda su un contesto condiviso

Una maggiore libertà ha senso solo se ciascuno dispone di ciò che gli permette di giudicare. Senza contesto condiviso, una decisione può sembrare corretta localmente ma contraddire un'esigenza, una scelta o un vincolo di portata più ampia.

Documentare non significa quindi accumulare resoconti. Significa rendere accessibili il problema da risolvere, l'obiettivo, le ipotesi, le ragioni delle decisioni, i limiti e le regole che stabiliscono quando coinvolgere altri responsabili. Mettere questi elementi per iscritto permette di agire senza dover cercare a ogni passaggio chi possiede le informazioni. Il contesto diventa uno strumento per decidere, per le persone come per gli agenti.

Qui cambia anche il ruolo dei Garanti. Non controllano ogni iniziativa: rendono comprensibili le regole, ne tutelano la coerenza e intervengono quando una decisione le supera. Senza meccanismi di protezione, la libertà può diventare caos; se tutto deve essere approvato dall'alto, rimane soltanto una promessa.

Anche il management si trasforma. Una parte del monitoraggio e del coordinamento può essere svolta in altri modi. Ma rimangono persone da accompagnare, esperienza da trasmettere, conflitti da risolvere e scelte difficili di cui assumersi la responsabilità. Queste responsabilità non scompaiono insieme alle attività che un tempo davano loro forma.

Fare di più ciò che conta

Sarà forte la tentazione di misurare questa trasformazione in base al numero di funzionalità prodotte o alle ore risparmiate. L'IA può effettivamente farci produrre di più. Ma riempire subito il tempo liberato con nuove attività significherebbe accelerare il vecchio modo di lavorare senza metterne in discussione lo scopo.

Possiamo anche usare questa capacità per capire meglio le esigenze, verificare le ipotesi, migliorare ciò che esiste e rinunciare a ciò che non crea valore. Nulla di questo avviene automaticamente. Bisogna scegliere dove investire l'attenzione e le risorse liberate.

Vivien, in Malorean Corp., non scopre soltanto che il lavoro di prima occupa meno tempo. Scopre di poter mettere la propria esperienza al servizio di un altro problema, senza garanzie di successo. Il progresso non consiste soltanto in ciò che abbiamo automatizzato, ma anche nelle possibilità aperte da questa trasformazione.

All'inizio della storia tutti si stupivano di quanto velocemente Lucas riuscisse a costruire. Alla fine la domanda è un'altra: perché costruire, per chi e con quali conseguenze?

La sfida dell'IA non è farci fare sempre più cose. È darci i mezzi per fare di più ciò che conta davvero.