

Con il fine di strutturare meglio le informazioni contenute all'interno dell'intervista, questa è stata divisa in capitoli. Ad ogni riga della tabella corrisponde un capitolo.

Nella colonna sinistra è riportata l'intervista, in quella di destra sono riportati il titolo del capitolo, che ne riassume il contenuto, e gli insights ricavati dal capitolo.

I colori mettono in relazione l'insight ricavato con il pezzo di intervista a cui fa riferimento. Per ogni capitolo, i colori fanno riferimento solo al capitolo specifico e sono indipendenti dagli altri.

<p>R come prima cosa volevo chiederti qual è stato il tuo ruolo all'interno del team di progetto e come contribuito al successo del progetto.</p> <p>I Come ruolo, quello di sviluppatore progetto full stack, quindi mi occupo sia la parte di back end che di front end L'unico in ufficio, poi c'è vabbè Oscar che è il Techincal Leader. Quindi niente come sviluppatore, il compito dello sviluppatore a ricevere task elaborati Dal Functional Designer e preanalizzati dal Leader tecnico Questi task vanno di nuovo rianalizzato dallo sviluppatore e poi messi in pratica, quindi ci va la scrittura di codice che poi viene messa in review che passerà in code review verso il leader tecnico</p> <p>R Però, comunque il team di progetto è sempre cambiato nel tempo. È vero che adesso Il team si è un po ridotto per l'ultima release, però nei periodi precedenti erano composti da altri membri che poi sono cambiati. Il team ha avuto varie vicissitudini.</p> <p>I Sì, quello sì. quando ho iniziato [parla del progetto precedente con lo stesso cliente, non del Partner Portal] c'era un solo altro sviluppatore quindi eravamo due sviluppatori Lì però, proprio nella stessa stanza dove ci trovavamo. Poi lui è andato via e sono rimasto solo più io e poi è subentrato Riccardo. E poi, quando è iniziato il progetto del portale, Quando è arrivata la conferma che Coolshop aveva vinto il progetto del portale, io sono andato sul Portale, lui è rimasto sull'altro applicativo [per l'AMS] Con me hanno iniziato altre tre figure dev, che sono state successivamente sostituite.</p>	<p>RUOLO</p>
<p>R volevo chiederti come reputi il passaggio delle informazioni all'interno del team di progetto e quindi se tu sei sempre conscio di quello che ci si aspetti che tu faccia e quando è previsto che questo venga finito.</p> <p>I Allora, Eh, diciamo che non è sempre così, nel senso che non sono sempre conscio di Quando devo consegnare diciamo il</p>	<p>COMUNICAZIONE INTERNA</p> <p>Mancanza passaggio informazione effort stimato</p> <p>Periodi concitati riducono qualità passaggio informazioni ed efficienza del team</p>

mio task, quanto tempo ho non è sempre così. Diciamo che va a periodi. Ci sono dei momenti in cui siamo più strutturati e quindi i task sono compilati particolarmente bene, c'è una data di consegna e ci sono delle assumption. è linkato un FD, quindi in quel caso lì effettivamente la comunicazione è buona. Viene lasciato poco al caso, diciamo così. Quindi lo sviluppatore sa esattamente cosa deve fare, in quanto tempo deve farlo. Poi ovvio che nei periodi un po' più concitati, eccetera, quando arrivano più cose ci sono dei cambiamenti eccetera magari si perde un po' questa costanza nel compilare i vari task, di allineare bene lo sviluppatore e quindi ogni tanto ci sono altre mancanze di informazioni, quindi si oscilla fra fra questi due stati

R

Tu noti che poi effettivamente la tua efficienza o quella del team si riduce nei periodi meno strutturati o non noti una grande differenza?

I

Sì, accade sicuramente

R

E sai indicarmi all'interno del progetto dei periodi in cui effettivamente hai notato queste differenze in cui c'è in cui questa efficienza si è ridotta?

I

Ma adesso non saprei fare un esempio. L'unico esempio che so fare è quello che è stato forse due settimane, quello più fresco. C'era da portare a termine un task, è stato elaborato meno questo task, quindi è stato elaborato meno dal Jira del cliente, quindi c'era copiata la parte scritta del cliente. Non era stata abbastanza elaborata, c'era un titolo ancora che faceva riferimento ad un'altra cosa, quindi lì ho perso tempo per capire cosa dovessi fare. Ovviamente poi ho parlato con il Technical Leader e le cose si sono messe a posto. Ora, non ho perso un'infinità di tempo, però cioè ho bisogno di quel passaggio che sarebbe stato, diciamo inutile se il task fosse stato completato subito nella corretta maniera.

R

L'esempio mi è molto chiaro, però, tu hai notato anche tipo TI vengono in mente delle mensilità, proprio dei periodi di tempo in cui effettivamente ciò è accaduto

I

Ma no, non te lo saprei dire per dire se c'è una appunto una regolarità o dei periodi specifici tali da tanto, cioè i task sono tanti, magari spesso ci sono ... Ecco, forse ogni tanto ci sono dei task più lunghi che impieghiamo più tempo per essere fatti

Collaborazione del Team  
positiva

Cambio di Priorità comunicato  
con efficienza

che hanno più sostanza, diciamo e quindi magari sono preparati con più accuratezza rispetto a quelli più Piccolini che si susseguono uno dietro l'altro e quindi magari Passano direttamente allo sviluppatore senza essere, diciamo elaborati sufficientemente, ecco, forse quello.

R

Ok, d'accordo, va benissimo. Parlando un po' della collaborazione con il team, se accade che magari stai sviluppando qualcosa che può avere impatti su ciò che sta sviluppando o che è stato sviluppato da un'altro membro del team, Vorrei chiederti un po come il team è riuscito a collaborare da questo punto di vista, se il passaggio di informazioni è stato corretto, se è stato facile per te capire subito quali potesse essere gli impatti.

I

Ma sì, dal mio punto di vista questo ha funzionato bene, nel senso che è capitato poco, magari nell'ultimo periodo ci siamo io e l'altro dev che lavora sull'AMS, può capitare che lavorassi su una cosa che stava già modificando lui, però anche prima, quando c'erano gli altri con cui lavoravo sul Portal Poteva capitare, però non sono stato nessun problema, diciamo che una Cosa importante è stato lo stand up E quindi sai esattamente su questa lavoravano gli altri, ogni tanto è capitato capire che qualcuno lavorasse sul back end di una cosa che poi poteva impattare su di me, E poi comunque c'è il leader tecnico che quando ti assegna un task te lo segnala se qualcuno sta lavorando su quello. Se non se non dovesse succedere questo appunto lo stand up della mattina funziona da questo punto di vista.

R

Ottimo. Ti è mai capitato che poi effettivamente di lavorare con altri membri del team con cui invece, per disattenzione o altro, non riuscivi a collaborare?

I

No, assolutamente. Ho collaborato con molti membri e non ho avuto mai problemi

R

Un'ultima domanda su questo ambito, se ti è mai capitato che la priorità di un task che stavi seguendo è stata cambiata, magari ne è subentrato uno con una priorità maggiore. Inoltre, se questo ti ha comportato dei problemi di efficienza, se è stato comunicato bene anche la motivazione per la quale la priorità era cambiata, Se è successo raramente o frequentemente.

I

<p>Allora capita. non saprei dire se frequentemente Però succede, succede abbastanza spesso. Eh problemi no problemi no. Per due motivi, uno perché sicuramente usiamo bene Jira, quindi nel momento in cui devi bloccarti su uno lo metti in pending e poi lo recuperi successivamente dalla dashboard, inoltre lo puoi commentare spiegando perché lo metti in quello stato lì. Altra cosa che aiuta questo discorso qua è di nuovo lo stand up perché confrontandoci giornalmente con il Technical Leader su cosa stai facendo, È ovvio che loro sono aggiornati, loro ti possono aggiornare sulle priorità, quindi si notifica un cambio di priorità e quindi il team è sempre allineato. Ogni tanto può capitare che un task rimanga in pending e ce lo si dimentica in quello stato. è capitato.</p>	
<p>R Ok, d'accordo. Vorrei farti delle domande invece sulla collaborazione con il technical leader, perché per come è impostata la metodologia si fa molto affidamento su di lui per le scelte tecniche, soprattutto di alto livello, si fa molto affidamento su di lui, è un po' lui che gestisce tutta l'architettura e che è un po' la motivazione per il quale poi è lui che vi rielabora un po' i task prima di consegnarvi perché vi dà un po' di scope. E quindi sì, volevo chiederti se ritieni che poi questo tipo di decisioni sia troppo centralizzato sul technical leader</p> <p>I Sì direi che è centralizzata. Si può fare forse una differenza fra quello che è il cuore vecchio dell'Applicativo. Mhm quindi il catalogo e configuratore e la parte nuove di Leads, orders e compagnia bella che sono state sviluppate tutte ex novo con un'architettura diciamo a microservizi no, quindi sono. Sono molto separate tra di loro le funzionalità. Utilizzano un'altro linguaggio back end. Hanno anche un data model loro separato, costruito ex novo, quindi molto Molto meno confusionario rispetto al vecchio. E quindi diciamo che mentre sul vecchio la conoscenza che ha il technical Leader. Del data model delle procedure, tutto quello che è stato fatto è fondamentale E su quello difficilmente uno sviluppatore nuovo riesca a inserirsi nel flusso, ci vuole esperienza, forse nella parte sviluppata ex novo, che è molto più leggera, molto Più smart, diciamo così, ecco forse uno sviluppatore tecnicamente potrebbe Fare anche delle scelte Diciamo di livello tecnico, senza dover per forza passare sotto il la supervisione del technical leader</p> <p>R Ma quindi capita?</p> <p>I Non accade perché comunque La metodologia è quella</p>	<p>RAPPORTO CON TECHNICAL LEADER</p> <p>Le decisioni dell'applicativo vecchio sono in mano al Technical Leader, ma per la nuova parte i developers hanno maggiore facilità a proporre le proprie soluzioni</p> <p>Distribuire le decisioni permette al Technical Leader di essere più libero</p> <p>Anche se il Technical Leader è sovra impegno, le risorse non sono mai disallocate</p> <p>Se il technical leader è sovra impegnato, Può capitare che le risorse siano bloccate nello sviluppo, ma nel caso passano ad altro senza perdere di efficienza</p>

sempre di passare dal Technical Leader, Si è quella sempre di Sottoporre al Technical Leader le soluzioni che uno propone. Diciamo che uno sviluppatore che non è il TL è più agevolato nel proporre soluzioni su quelle parti da applicativo nuove quindi sono un po più semplici rispetto al codice Legacy, che effettivamente è un po più complicato eccetera, quindi l'esperienza del TL la conoscenza è fondamentale.

R

Ok, però pensi che il team ne gioverebbe se le decisioni fossero più diffuse nella parte nuova?

I

Ah se ne gioverebbe... Allora è difficile da rispondere a queste domande, ovvio che se la decisione passa attraverso uno, la decisione è più rapida, è più veloce . Ovvio che più condivisione c'è più il team è informato e magari questo è un vantaggio che poi ritorna utile nel futuro È che i vari sviluppatori in questo modo Acquisiscono una certa autonomia che può tornare utile.

R

Con ciò mi collego anche alla domanda successiva, perché effettivamente, sempre per come è impostata la metodologia il TL è diciamo pieno di cose da fare, perché ovviamente fa code review, a volte sviluppa, Si occupa di diprendere decisioni di alto livello, quindi immagino che in teoria le la possibilità di prendere decisioni, magari anche di più basso livello, senza passare dal TL, possa Prima di tutto snellire il suo compito

I

Per quanto mi riguarda è avvenuto appunto in quella parte di applicativo Nuove.

Penso. Ad esempio alla warranty Che ho imbastito io sulla base dell'FD. Però partendo da zero e quindi mi sono creato io il data model eccetera Ho avuto un'autonomia che Difficilmente avrei potuto avere magari nella parte Legacy di applicativo.

R

Ok, molto chiaro, però mi chiedevo anche che voi DEV, ma anche il technical leader, Capita che quindi prendiate delle decisioni che proprio non sono documentate perché sono Non all'interno del functional design o mi sbaglio?

I

Oh, diciamo che queste decisioni non sono all'interno del functional design perchè questo è a livello superiore. Nel senso, le functional designer mi dice come deve funzionare quella funzionalità. Cosa devo mostrare all'utente, per

<p>esempio. Poi, per esempio, il data model è una cosa che decide lo sviluppatore e non c'è nel functional design. Ci sono scritte delle proprietà di una caratteristica, un oggetto, una funzionalità o un qualcosa. Il data model è documentato direttamente nel codice o a database Con i tipi, eccetera. Noi siamo questo livello di decisionale.</p> <p>R</p> <p>Sempre riguardo il technical leader che effettivamente ha un sacco di compiti sul groppone. Mi chiedevo se, siccome La sua attività di decisione di passaggio delle informazioni verso di voi, verso voi developers vi sblocca la possibilità di iniziare un determinato task, hai mai notato durante il progetto che quella situazione fosse bloccante per voi, Nel senso che banalmente non avevi task da sviluppare perché il TL era impegnato in altro</p> <p>I</p> <p>Può capitare questa cosa qua, ma molto raramente. Io sono stato scarico veramente poco nell'ultimo anno, quindi effettivamente non è che capita molto spesso. Altro discorso è quando ci sono delle cose bloccanti. Di nuovo c'è bisogno del technical leader. Lì è successo, è successo più volte, però non si è mai, Non c'è mai fatto dell'inefficienza perché diciamo che c'è sempre stato un'altro task c'è sempre stato qualcos'altro da fare. Ecco, non non c'è mai stato un'efficienza da questo punto di vista</p> <p>R</p> <p>OK, quindi banalmente, una volta che si è bloccati lo segnali e poi giri su un'altro task dove magari hai già da fare</p> <p>I</p> <p>Esatto, lo metti in Pending e poi ne parli durante lo stand up meeting.</p>	
<p>R</p> <p>Ok, chiaro, molto chiaro. Andando avanti noi dall'Agile abbiamo preso un momento, un evento che si chiama Retrospective con il quale il team si riunisce e ognuno porta sul tavolo dei punti in cui vorrebbe che il team migliorasse o del delle idee da portare avanti. So che ci sono stati questi momenti durante la vita del progetto. Tu come li hai trovati? Li hai trovati utili, li hai trovati superflui?</p> <p>I</p> <p>Allora sicuramente il momento di confronto è utile. Non vedo come possa non essere utile, è vero che adesso, ripensandoci, l'ultima retrospective che abbiamo fatto è passato molto tempo, adesso non ricordo esattamente. Però effettivamente di</p>	<p>MIGLIORAMENTO DEL TEAM</p> <p>Retrospective utili, ma devono essere seguite da fatti</p> <p>1-to-1 sia con TL che con Paladin molto utili, in quanto meno espansivi</p>

quella retrospective non ricordo molto di cosa sia stato detto e credo che poi le cose che sono uscite Non sono state, diciamo che le cose che sono venute fuori non sono state fatte seguire delle dei fatti non sono stati presi accorgimenti, sono rimasto un po lì. effettivamente si sono persi

R

Per quello che ti puoi ricordare riguardo a ciò che è uscito, magari quello che avevi portato, tu pensi che le idee che siano state portate sul tavolo erano utili e pensi che il team ne avrebbe potuto giovare.

I

Io mi ricordo che sicuramente erano utili. Tante cose utili, però poi è mancata la metodica, la continuità e quindi magari nelle due settimane successive si è tentato Di portare delle soluzioni a quelle cose che erano venute fuori durante la retrospective, ma poi si son perse le cose. Quindi ecco la continuità non ha consentito di di portarla a quelle cose.

R

E in genere però, trovi che il team andando avanti con il progetto, abbia imparato da sé sia migliorato sia diventato più efficiente anche rispetto alla comunicazione tra i membri o con il technical leader o il paladin.

I

Non saprei perché rispetto a prima molto [progetto precedente] rispetto a prima. è stato proprio un'altra, un'altro paio di maniche, da quando mi ha iniziato il portal, Cioè da un anno a questa parte, Sì, abbiamo forse migliorato sì Singolarmente alcuni meccanismi, io come sviluppatore sono migliorato eccetera però non saprei individuare dei flussi che sono nettamente migliorati in un anno. Penso che funzionavamo già abbastanza bene prima

R

Ok, però riesci a pensare a trovare una motivazione per il quale poi il team abbia smesso di un po di migliorare.

I

diciamo che un miglioramento c'è stato, Non c'è stato quel miglioramento che c'è stato nel progetto precedente Non saprei, su questo non saprei dirti perché.

R

Però se ti viene qualcosa in mente mi fa piacere, però va bene, va bene così. Insomma, ma in realtà questa tua non risposta potrebbe essere in sé una risposta puramente perché magari Se si fossero fatti delle retrospective, magari più strutturate, magari portandole avanti ci si poteva anche

<p>chiedere, ma perché rispetto a prima non non stiamo migliorando così tanto? Cosa ci blocca?</p> <p>I Ah, quello effettivamente potrebbe una risposta, perché le retrospettive le abbiamo interrotto dopo quattro mesi.</p> <p>R E nel nel progetto precedente facevate le prospettive oppure no?</p> <p>I No, no, facciamo le Retrospettive. Una cosa che abbiamo sempre fatto sono i 1-to-1. Quello si è sempre fatto e quello funziona abbastanza perché poi le cose che ci diciamo tra sviluppatore e TL. Poi vediamo che il TL prende dei provvedimenti o appunto Magari lo facciamo noi. Lì è un discorso uno a uno e quindi è più semplice, mentre nella retrospettiva, vengono fuori tante idee, essendo tanti che mettono le proprie idee e quindi se non c'è un metodo le cose si perdono.</p> <p>R Ok e pensi che il One to One, oltre che con il TL, potrebbe essere utile se è fatto anche con il paladin del team o meno?</p> <p>I Sì, secondo me è utile.</p> <p>R Perché?</p> <p>I Ma perché credo che Gli sviluppatori possono dare spunti. E per esempio adesso mi viene un esempio sulla compilazione dei task sulla creazione dei test case eccetera e viceversa. E anche il paladin, può dare degli spunti allo sviluppatore su come per esempio, interfacciarsi appunto col Paladin o come utilizzare al meglio Jira. Secondo me il confronto può essere solo utile.</p>	
<p>R Ok, d'accordo, benissimo. Cambio argomento o una cosa che di cui abbiamo parlato un po' prima, di cui è uscito fuori, era il discorso che a un certo punto c'è stata un cambio nella figura del Paladin Senior. Intorno alla release 5. Volevo chiederti se secondo te questo passaggio di consegna ha portato inefficienza all'interno del team, se il team ha risposto bene.</p> <p>I</p>	<p>PASSAGGIO DI CONSEGNA</p> <p>Passaggio valutato positivamente</p>



<p>Ma secondo me no non ha portato nessuna inefficienza e devo dire è andato molto bene.</p> <p>R</p> <p>E come Stile di project management? anche solo di scrittura dei task?</p> <p>I</p> <p>No, quello no, no. Poi magari noi sviluppatori ce ne accorgiamo meno, in quanto ci arriva arriva il task già filtrato dal diciamo confronto fra i Functional designeR e Technical leader, quindi magari il passaggio di consegne ha impattato più su TL che prima ha a che fare con una persona e poi magari con un'altra e quindi magari Ci sono stati dei momenti in cui il nuovo Paladin senior, alcune cose doveva ancora elaborare completamente, appena entrato sul progetto, mentre noi interfacciandoci già su qualcosa che magari già elaborato dal TL più delle volte ci siamo accorti meno di questo passaggio. Non Abbiamo sentito questo passaggio.</p> <p>R</p> <p>Per quanto riguarda le comunicazioni? Magari un cambio di priorità tra due task che comporta il passaggio da uno all'altro?</p> <p>I</p> <p>Neanche, Di nuovo, forse ha aiutato anche lo stand up giornaliero, anche quello potrebbe aver aiutato il nuovo Paladin nei primi moment a non perdersi nulla. Quindi può darsi che quello ha agevolato il passaggio di consegna.</p>	
<p>R</p> <p>Volevo chiederti se hai mai avuto difficoltà durante il progetto e nel caso come hai risolto, come sei Stato aiutato a risolvere quelle difficoltà?</p> <p>I</p> <p>difficoltà si momenti di difficoltà si. Uno, forse a Febbraio dello scorso anno. È stata l'implementazione di una All'interno dell'applicativo di una libreria per la gestione delle delle tabelle In quel periodo era Subentrato il secondo TL a dare una mano. Quindi io ero sotto di lui e ho implementato questa libreria ed effettivamente È stato un mese abbastanza duro, coi tempi abbastanza contingentati, perché Infatti, di diciamo che Ho lavorato parecchio, forse anche perché la stima che mi aveva dato il TL e mi ero dato anch'io, Alla fine si è rivelata una stima errata per le mie capacità e quindi quando le stime sono errate, provi a starci dentro, è ovvio che non si lavora benissimo, non è semplice lavorare, però poi Alla fine, si è arrivati al fondo all'implementazione di quella libreria, diciamo.</p>	<p>Difficoltà a causa di stima sbagliata Technical Leader aggiuntivo</p> <p>Si è riusciti a tenere flessibile la tabella di marcia</p> <p>Stima secondo TL magari meno precisa del primo, ma necessaria per non bloccare i lavori</p> <p>Utilizzo di nuove tecnologie valutato positivamente – pochi rischi, developer contento</p> <p>Compilazione dei task da migliorare</p>

R

però mi chiedo se l'azienda in qualche modo TI è riuscita ad aiutarti. Aveva degli strumenti che il TL ha potuto fornire per aiutarti. Non so, una mano.

I

Ma in verità l'aiuto, in quel caso lì è stato non dato nell'aiuto fisico, del TL che si è messo a correggere parti del codice eccetera, ma capire che forse la stima era proprio contingentata e quindi Consentire, agevolare un tempo più lungo per portare a termine questo task, non sentire troppo la pressione. Dopo con il TL ci siamo confrontati eccetera su alcune cose, non è che sono stato abbandonato, quello no. Quindi magari anziché lavorare su altre cose, si sono gestiti vari task in maniera tale da lasciarmi il tempo a sufficienza per portarlo a termine. Ci si organizzati diversamente rispetto a quello che era la tabella di marcia.

R

E poi mi chiedevo Se però reputi che se quella stima tu l'avessi fatta solo con il primo TL perché banalmente ha più le mani sul progetto, se la stima in quel caso sarebbe stata comunque sbagliato lo stesso, oppure se effettivamente avrebbe avuto un po più di occhio.

I

Ma, Può darsi che lui, conoscendo meglio il progetto, avremmo fatto magari una stima più un po più corretta, semplicemente perché lui conosce la parte legacy dell'applicativo, quindi dove sono già state implementate tutte le tabelle che verranno modificate dalla nuova libreria e quindi forse avremmo fatto una stima un po più verosimile. Quello sicuramente però è stato un periodo in cui lui Era già oberato di cose, appunto, per quello era subentrato il secondo TL, per dividere alcune cose. adesso non ricordo esattamente, ma credo che fosse il periodo in cui doveva fare diverse stime. Quindi eravamo arrivati a questa organizzazione in cui io e un altro lavoravamo sotto il nuovo TL e facevamo riferimento solo a lui e non disturbavamo l'altro, in modo che non togliessimo tempo.

R

Ok, chiarissimo, chiarissimo e Diciamo, questa è anche la parte di progetto che ti è piaciuta di meno?

I

Sì, non è che non mi sia piaciuta Eh, forse è stata la più difficoltosa, poi No, non mi è dispiaciuta perché appunto si trattava di utilizzare La libreria di React, quindi tecnologia nuova e lavorare su tecnologie nuove, almeno per quanto mi

riguarda, è sempre piacevole, cioè a me piace sempre. Quindi preferisco sempre lavorarci. La parte nuova rispetto a quella vecchia.

R

E il progetto VI ha dato possibilità di sfruttare delle tecnologie nuove o si è scelto di andare un po più sul sicuro?

I

Assolutamente, c'è stato un cambio drastico, il Portal ha permesso di introdurre ReactJS, che rispetto AngularJS è una tecnologia sicuramente più attuale.

R

In genere tu pensi che l'utilizzo di queste tecnologie possa essere Più rischioso all'interno del progetto, nel senso che magari la stima può A consuntivo, essere più variabile? Cioè magari è un po più difficile rientrare all'interno della stima oppure no?

I

No, non credo. Non credo poi forse questa domanda vale anche da sviluppatore a sviluppatore magari io mi sono trovato meglio con le tecnologie nuove, rispetto quelle prima, magari sono anche un po' più confidente nelle stime rispetto a tecnologie vecchie come Angular. Poi magari altri lo sono meno. Diciamo che non c'è qualcosa di imputabile alla tecnologia in sé.

R

Invece la parte del progetti che ti è piaciuta di più?

I

E di nuovo queste parti qua, quelle strutturate a microservizi . Dove ogni volta partitivi da zero, Data model nuovo, API from scratch, anche perché partendo da zero Come dicevo prima, c'era più autonomia decisionale. Sicuramente più interessante anche da sviluppare

R

E quindi ti piacerebbe, magari in un progetto futuro, avere più autonomia decisionale?

I

Sicuramente, più che autonomia decisionale è iniziare qualcosa da zero. In questo caso queste parti nuove sono state create da zero e quindi necessariamente quando crei qualcosa da zero. Le scelte le fai tu Mentre quando vai a creare qualcosa del genere con una parte Legacy, devi seguire dei binari, puoi prendere meno decisioni.

R

Certo, molto chiaro. volevo chiederti se qui o in altre aziende hai utilizzato altre metodologie di project management quindi altri processi.

I

L'ho scoperto praticamente solo in coolshop, nell'azienda dove lavoravo prima non avevamo un Giusto flusso codificato, eravamo più ruspanti.

R

Ti trovi al 100% meglio con la metodologia di Coolshop o vorresti ripendere qualcosa dal vecchio modo di lavorare?

I

No, risposta secca. Prima non c'era proprio metodologia. Non c'era molto questa metodica

R

In generale, cosa pensi Si potrebbe fare meglio? Cosa si potrebbe fare meglio come processi, come condivisione delle informazioni rispetto la metodologia che utilizziamo.

I

Ma sicuramente forse la compilazione dei task, è un po il tallone d'Achille. Quello che dicevo all'inizio. Spesso sono compilati alla perfezione, soprattutto quando i task da fare sono sono grossi, quindi c'è molto materiale, Sono preparati bene, magari quelli che sembrano un po più semplici, un po più scontati, magari sono un po trascurati, ci Si perde un po di tempo che si potrebbe non perdere.

R

Eh, mi spieghi cosa, Come definisci un task preparato bene.

I

Allora un task preparato bene dipende, se è un bug il test case Elencato bene, pulito, quando è avvenuto, in che ambiente, elenco puntato di come faccio ad ottenerlo. Quello sicuramente è ben fatto, Le altre cose delle taskche che deve essere fondamentale è Cosa voglio ottenere alla fine. Sembra una banalità, ma ogni tanto ci sono alcuni task dove alla fine non è specificato che cosa vogliamo ottenere anche a livello grafico di interfaccia, si danno per scontato delle cose, può capitare. Si possono creare dei fraintendimenti

R

Quindi è un po il passaggio dell'informazioni da parte funzionale a parte DEV.

I

Sì, esatto Diciamo che dovrebbero esserci Tutti gli elementi compilati, qual è la situazione attuale, Come siamo messi, cosa vogliamo ottenere E suggerimenti del technical leader per passare dallo stato attuale a quello sperato

R

Sì, ecco ti volevo chiedere se effettivamente c'è un momento in cui tu hai la possibilità di dire al funzionale che ti ha scritto in task, guarda che mi mancano queste informazioni Affinché questo sia un buon task, insomma, quindi una sorta di coordinamento.

I

Sì, sì, quello quello c'è. L'ultima volta che è successo ho detto al TL che mancano queste cose qua, E poi anche lui si è reso conto che qualcosa è sfuggito. Qualcosa mancava, però problemi di comunicazione non ne abbiamo

R

No, perché poi stavo anche pensando al fatto che potrebbe... Cioè, è un punto che, ad esempio tu potresti portare in un ipotetico One to One dev paladin.

I

Ah, quello sicuramente.

R

anche perché lato paladins si fanno i test case per il cliente nel momento in cui lo sviluppo è già finito, quindi magari potrebbe essere una buona idea fare quello prima di iniziare lo sviluppo, Banalmente, perché può aiutare lo sviluppatore Nel capire esattamente quello che va fatto, qual è il processo che deve essere implementato?

I

Magari ciò che potrebbe fare uno sviluppatore è buttare giù uno scheletro di compilazione della descrizione del task In caso di bug o di sviluppo di feature. Che il TL o il Paladin potrebbero seguire o da cui potrebbero prendere spunto. Poi si modificheranno assieme Qualcosa che possa dire "questo è il mio task ideale" Questo potrebbe essere anche ovviamente un'idea di argomento da portare in un One to one, in un confronto.