

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.

R	RUOLO
<p>Allora prima cosa vorrei chiederti, qual è il tuo ruolo all'interno del team di progetto e come contribuito al successo del progetto?</p> <p>I</p> <p>il mio ruolo all'interno del team progetto è il technical leader, In Coolshop questa figura viene chiamata Hero. E la mansione riguarda la coordinazione. Della parte di sviluppo e la Quality ensurance, quindi comunque garantire che il prodotto finale abbia una certa qualità, un risultato buono e soddisfacente per il cliente, ma anche che sia nei tempi, quindi si tratta di appunto una mansione anche organizzativa, che permette di dettare tempi di sviluppo E cercare di arrivare al gol del risultato di rilascio del progetto, in questo caso verso il cliente nei termini stabiliti. Come ho contribuito al successo? Supportando il team perché Come elemento laterale si ha che non ho soltanto il compito di verificare la qualità, il risultato, la l'attinenza poi a quello che è stato appunto scritto, magari negli FD, quindi l'attendibilità della soluzione tecnica sviluppata, ma anche quella di supporto al team, quindi ovviamente Una buona parte del mio tempo viene anche occupata nell'analisi delle richieste, quindi appunto problematiche tecniche, come risolverle e a volte a supporto Del team, quindi, se un elemento, uno sviluppatore, un elemento del team sia sviluppatore che in realtà anche la parte funzionale quella paladin, si ha necessità di analizzare, capire, uscire da una situazione di stallo dovuto a un problema tecnico o analitico, perché comunque È importante Far emergere Problemi che si potrebbero avere in futuro Applicando una certa soluzione. Quindi il mio compito di supportare il team e quindi di risolvere i problemi in caso di stallo Sono dovuti a OK, ma questi elementi che dobbiamo andare a sviluppare, che dobbiamo andare a gestire, hanno delle interconnessioni? Ci possono essere delle problematiche, se questa roba viene sviluppata in un certo modo, può avere dei conflitti con Elementi già sviluppati o ancora peggio, elementi da sviluppare e quindi tenere le fila, coordinare Tutto questo Permette a me di avere una buona riuscita nel progetto. Quindi avere un overview generale di tutto e poter supportare il team Nella interconnessione tra loro. quindi, appunto, la parte funzionale che va a fare un'analisi. Io dovrò fornire una stima e quindi dovrò cercare di rimanere nei tempi e questo è anche un'altro elemento che concorre al risultato del progetto, perché ovviamente se non si sta nei tempi Non si raggiunge il gol del risultato e quindi appunto dicevo, aiutare la parte funzionale a ad avere delle stime, ad avere un'analisi completa e coordinare le varie</p>	

<p>analisi e i vari sviluppi insieme per poter prevenire, prevedere e non avere delle situazioni da risolvere a posteriori. E che portano, diciamo a una situazione Di ritardo perché ovviamente se possiamo Preventivamente verificare dei conflitti che in un progetto molto lungo, come è stato questo, si possono verificare Nel corso del progetto si potrà avere un risultato migliore e una un'efficacia migliore</p>	
<p>R Quindi, parlando della comunicazione che hai con il membro paladin del TEAM, Ti è sempre chiaro quali sono i requisiti richiesti dal cliente per una certa funzionalità? O magari in cosa consiste un bug che il cliente ha segnalato o questa comunicazione a volte risulta inefficace?</p> <p>I La comunicazione è molto importante, L'unico problema che si ha quando c'è un intermediario, il paladin è l'interpretazione della richiesta. I problemi principali riscontrati sono nell'aver cambiato dall'inizio del progetto, quindi da dai vari sviluppi del progetto. Non intendo questo, ma anche quello. Prima precedente a questo che è un proseguo, diciamo, è un un'evoluzione un upgrade del progetto stesso, aver capito tante figure di riferimento [si riferisce al vecchio progetto con lo stesso cliente] E aver cambiato tanti figure di riferimento ha portato secondo me a delle situazioni di Non si ha un overview completa, magari del progetto e quindi ritornando al discorso dell'efficacia della comunicazione, E di un intermediario tra il cliente e la parte Hero, quindi il paladin, come gioco del del telefono senza fili, sic hahe la comunicazione Pian piano possa essere degradata, quindi posso avere dei lag, delle mancanze di informazione. Questo perché anche Dalla sorgente, quindi dalla parte cliente si danno per scontato una serie di cose, di informazione che viene data dal paladin dà per scontato altre volte. Quindi a volte Leggendo La richiesta iniziale del cliente ho potuto Ricontrare un'analisi critica diversa da quella che mi è stata presentata successivamente. Perché vado a connettere più informazioni, dato il mio background e quindi Unisco più puntini e tiro fuori, magari la problematica che sta da tutt'altra parte. E non dalla problematica che mi è stata invece posta da risolvere. Quindi il punto chiave è Arrivare a fare analisi e dire che il problema sta in una certa zona deve essere risolto È importante, è un compito del paladin, a volte diventa però difficile fare un'analisi completa e chiara e avendo un'analisi che è un risultato della richiesta del cliente, si perde l'informazione principale del cliente, quindi si può avere una perdita di informazioni e questo perdita di informazioni porta ad avere. Magari un approccio della Descrizione del problema, dell'aria di risoluzione del problema E non coincide al 100% con il problema stesso, perché magari, appunto l'informazione è degradata, l'informazione mancante che è arrivata dal cliente Porta ad avere un un'analisi Per la risoluzione del problema Incompleta o comunque parziale, che Potrebbe mancare di appunto quelle interconnessioni</p>	<p>COMUNICAZIONE</p> <p>Mancanza di overview dei Paladin a causa di Turnover</p> <p>Paladin Come intermediario potrebbe provocare perdita di informazioni</p> <p>Migliorare descrizione dei task</p>

di cui parlavo prima e quindi Del "attenzione ci sono degli dei side Effect degli effetti collaterali" tra una certa problematica e un'altra e quindi Potrebbe essere importante secondo me avere un flusso a più step nel quale c'è una descrizione più oggettiva del problema data dal cliente e una possibile analisi interna e questo dà un quadro completo a chi legge di Possibili altri Elementi che magari si possono perdere nell'avere soltanto la parte di analisi, senza avere il principale problema, cioè la descrizione iniziale del problema. Potrebbe portare a un processo di risoluzione di quell'analisi che in realtà non non è completamente al 100% buono

R

Mi è tutto molto chiaro, tranne il modo in cui miglioreresti questo aspetto.

I

Allora il modo in cui migliorerà questo aspetto sono due Qui mi stavo riferendo principalmente a risoluzione di bug, perché sulla parte di Raccolta del requisito, ovviamente la parte di paladin deve raccogliere requisito e una volta che lo raccoglie deve analizzarlo con la parte tecnica in modo da Fare questa analisi a due mani e quindi qui il problema, tra virgolette, è quasi risolto, nel senso che due figure che hanno il 100% delle informazioni del progetto. Possono insieme fare un'analisi e quindi eventuali elementi che si sono persi si possono Gestire in in tempi successivi, ovviamente si spera che entrambi abbiano al 100% se può sfuggire qualcosa, poi si va a vedere negli elementi successivi. In una situazione in cui invece manca questo elemento Di confronto tra Hero e Palladio, quindi la parte tecnica che dovrebbe avere background massimo del progetto è la parte funzionale che dovrebbe avere appunto la massima conoscenza dei flussi di funzionamento dell'APPLICATIVO, lì quello che Andrei a migliorare sono le informazioni e un quadro completo delle informazioni che devono arrivare a chi sviluppa, quindi cercare di Ovviamente, rendendo consci le persone che leggono, quella informazioni una parte iniziale in cui si ha un Ricapitolo delle informazioni ottenute dal cliente, un'altra sezione, un'altro capitolo in cui si hanno si ha l'analisi Della parte Paladin di questo caso parte funzionale per la risoluzione del problema e 1un terzo step che dovrebbe essere quello dello sviluppatore che leggendo Da dove si è partiti e qual è stata l'analisi fatta possa avere un quadro completo e identificare se ci sono state delle eventuali mancanze Praticamente quello che è successo a me è stato che da un'analisi Letta io poi andavo a chiedere, ma scusami, cosa ha scritto il cliente? E da lì andavo a raccogliere altre informazioni che erano ovviamente omesse nell'analisi. Perché l'analisi diceva, questa è la soluzione del problema, senza specificare bene il problema, quindi per me Un task chiaro dovrebbe avere una descrizione del problema più analitica e magari con le informazioni che ha dato al cliente. Così si ha l'informazione originale, più eventuali aggiunte, perché se sono state richieste. Ma qual è l'utente che ha questa problematica? Ma

<p>quali sono le informazioni del mercato che dove si ha questa problematica, ovviamente sono elementi da aggiungere. E poi una parte di analisi, dato questo problema con queste informazioni per risolverlo bisognerebbe andare a nascondere questo elemento. Lo sviluppatore lo legge dice sì, ma se ti nascondo questo elemento, sappi che quando l'utente è collegato in quell'altro modo, poi non ha modo di uscire da questa pagina e si trova in una situazione di stallo. Devo procedere con questa analisi? Avendo più informazioni, ovviamente secondo me si può avere un quadro completo che a volte è quello che mi viene chiesto da chi sviluppa? Ho bisogno di un'overview più generale, che spesso è molto complessa, Ha uno spettro molto ampio e quindi viene dato lo scope chiuso per risolvere quel task che però rende un po' ciechi le persone che vanno a sviluppare perché non sanno bene cosa stanno facendo e perché. E invece, È una cosa importante, secondo me da fare. Sensibilizzare e rendere coscienti le persone su Cosa stanno facendo e perché lo stanno facendo.</p> <p>R Ok quindi Inserire nei task lo scopo più ampio</p> <p>I Esatto</p>	
<p>R Prima hai delineato come uno dei problemi di comunicazione con il lato paladin e che c'è stato, questo cambio di personalità all'interno del team prima di tutto Come valuti che sia avvenuto quel cambio e come pensi che sarebbe potuto avvenire in modo più efficiente?</p> <p>I Più che efficiente, efficace. il concetto fondamentale qui è il Mi ricordo a quello che dicevo prima, il dare per scontato. Quindi avere un'analisi che manca di informazioni, cioè quando noi andiamo a dare un sunto a fare un KT un trasferimento di conoscenze. Verso un'altra persona, ovviamente devi Comprimere, devi avere un risultato più a punti, un summary Di quello che tu sai, conosci. E ovviamente quando vai a trasferire le tue conoscenze un'altra persona L'importante è un affiancamento. Il problema dell'affiancamento è che in un tempo limitato non andrai a vedere tutte le casistiche barra le casistiche passate Si spera che non ricapitino, quindi hai un tot di informazioni che ti vengono date dall'affiancamento, cosa buona e giusta perché è importante capire vedere come si lavora. In maniera proprio fisica con l'altra persona di cui dovrai andare a ricoprire il ruolo, ma L'altra problematica è le informazioni che vengono trasferite, condivise. E ovviamente è difficile che uno ricordi tutto o barra che possa condividere tutto quello che sa perché devi Andare a fare un sunto di di tutte le informazioni, quindi recuperare le informazioni principali da condividere e trasferire Per appunto fare questa questo passaggio di conoscenza Quindi fisicamente non è possibile trasferire Tutto il background, tutto lo storico Di conoscenza che si è avuto su quel</p>	<p>PASSAGGIO DI CONSEGNA</p> <p>Bisognerebbe cambiare approccio al passaggio di consegna</p> <p>Perdita di efficienza dovuto a turnover è fisiologico</p> <p>Vantaggi nel Turnover dovuti a conoscenza aggiunta al team</p> <p>Anche Turnover dev porta inefficienza fisiologica</p>

progetto A un'altra persona. E non so neanche Dirti se Aver letto tutti i documenti funzionali, di tutte le funzionalità presenti all'interno del progetto Possono rendere consci l'altra persona di tutto quello che fa il progetto Perché ci sono tanti piccoli cavilli, tanti, tante piccole cose che magari spuntano, diventano delle postille, diventano delle piccole CR dei cambiamenti che poi non vengono riportati nei documenti funzionali e sarebbe impensabile dover poi ogni volta risistemare e avere tanti documenti funzionali, Tante modifiche, tante cose per tenere traccia di tutto e quindi avere poi l'ultima versione. Sarebbe bello però molto difficile da starci dietro, anche molto oneroso, molto costoso. Quindi non ho una ricetta per poter dire sì il modo giusto per migliorare questo processo di trasferimento delle conoscenze e questo. Sarebbe bellissimo, appunto migliorarne l'efficacia trasferendo il numero massimo di informazioni. Il problema vero è che quando vai a informare un'altra persona di un tot di cose, Queste sono tante. Diventano molte, diventano difficilmente digeribili, diventano Dei punti non collegati l'uno tra con l'altro e così si ha alla fine della fiera? Tanti punti e se il nostro cervello ha tanti punti sparsi non connessi, ovviamente non verranno ricordati. Quindi l'unico modo che noi abbiamo per crearci delle mappe mentali che ci possono portare a ricordare e ad avere L'informazione completa e con lettere e puntini, quindi collegare e avere dei flussi che ci permettono di avere un'idea chiara di quali sono i processi. Quindi forse, Quello che sarebbe bello e avere dei documenti parlando appunto di unire i puntini, di avere un concetto di flusso, avere dei documenti più schematici, meno verbosi, perché alla fine si leggere tanto è giusto, ma ci rimane molto più impresso e diventa molto più facile spiegare a qualcun altro, avendo un supporto visivo.

Avere dei documenti in cui si possono avere dei flussi di funzionamento Delle varie, appunto, funzionalità dell'APPLICATIVO. Quindi, ad esempio, stiamo parlando in questo caso di un progetto che CPQ, il flusso di configurazione e aggiunta di un prodotto a una Quotazione, un preventivo. Qual è il flusso? Non c'è n'è uno solo, c'è ne sono N. Quindi andare ad avere uno schemino in cui dica è possibile accedere a questo, a questo flusso da La ricerca da un preventivo stesso, da la lista dei prodotti Da la lista dei prodotti preconfigurati e quindi, essendo preconfigurati ho questa postilla per cui con una configurazione già attiva, posso accedere direttamente o passare da un processo di carrello, quindi avere un flusso che ti declina tutte le possibili Entry point che vanno a Ottenere quel risultato già può essere un elemento che aiuta visivamente a dire Ah OK, quindi non è solo questo il flusso c'è ne sono N quindi diventa fondamentalmente un grafo con ramificazione molto, molto complicata che però magari cercando di recuperarlo a pezzi comunque avendo Uno schema di tutte le funzionalità che portano a quella A quell'attività Rendono conscio chi sta Entrando nel progetto, quali sono le Potenzialità di quel progetto, perché a volte Non si ha un quadro chiaro di come fare, come arrivare a ottenere delle funzionalità e tutti i punti di ingresso, appunto, che possono portare a quella funzionalità. Poi, ovviamente

avere e tenere aggiornato un glossario Cos'è un preconfigurato, cos'è un configurabile, cos'è una preconfigurazione, cos'è un prodotto, cos'è un Ghost? Già solo avere un glossario che dica tutte queste parolone che poi hanno un'accezione un significato intrinseco all'interno del progetto che quando normalmente si parla si dice sì perché quel prodotto configurabile ha al suo interno un Ghost Ovviamente il dire ha un Ghost Ha un significato molto forte e vuol dire che c'è un prodotto che si è nascosto all'interno di tutta la configurazione, ma ha anche delle altre accezioni. Quindi come vengono gestiti i prezzi, come vengono gestite Le interazioni degli altri elementi all'interno della quotazione all'interno di quella configurazione, dato che c'è un prodotto che deve essere trattato in maniera particolare. Quindi normalmente Day by day si va a dire abbiamo un problema col Ghost, io so Che quello vuol dire tante cose, tante interconnessioni, tante problematiche, tante piccoli corner case che possono Andare A inficiare quella quella funzionalità e quindi avere un'idea ben chiara dei punti da toccare per verificare quella problematica, quello sviluppo correlato. Ma magari un'altra persona non, sa O nel KT che è stato fatto Tra i paladin quella parte è stata omessa perché non è mai capitata una situazione Che ha fatto emergere quella funzionalità. Quindi sicuramente avere un glossario con una definizione dei vari termini utilizzati all'interno del del progetto può aiutare. E la fatica è ovviamente di tenerlo aggiornato e quindi ci ci vorrebbe dell'impegno da parte di tutti. Dei flussi, quindi dei documenti in cui si hanno dei diagrammi in cui spiegano i vari flussi
Può essere utile e poi si ovviamente andare a leggere i documenti funzionali precedenti. Questo può essere ovviamente d'aiuto perché si ha una parte magari Più descrittiva delle varie funzionalità. Però, appunto, come dicevo non viene, non vengono Aggiornati, ovviamente, perché appunto un documento funzionale viene aperto e chiuso, quindi non non abbiamo degli elementi, dei flussi all'interno della Dell'azienda che permettono, ma in generale dei flussi che permettono di aggiornare un FD se ci sono dei piccoli cambiamenti, delle piccoli CR, quindi bisognerebbe trovare 1 1 documento. Più snello, facile da tenere aggiornato Che quindi, appunto possa riflettere lo stato attuale di funzionamento dell'Applicativo E per me potrebbero essere dei appunto dei flussi, dei diagrammi che che vanno A evidenziare Gli attori del flusso, il flusso stesso e quindi in questo modo andare a elencare le funzionalità dell'APPLICATIVO.

R

Credi che questa soluzione Potrebbe essere d'aiuto anche nel caso in cui possa avvenire un turnover del technical leader nel progetto.

I

utile sì, Non sufficiente. Perché ovviamente L'altro elemento a supporto dell'affiancamento secondo me è Importante Il 100% di non perdere le informazioni, Quindi il gol di avere il 100% delle informazioni trasferite Secondo me deve trovare una soluzione

tecnica fantascientifica di un cerchietto da mettere in testa e trasferire tutte le proprie conoscenze a Un'altra persona. Quindi, per me è molto difficile che si possa arrivare ad un raggiungere 100% di trasferimento delle proprie conoscenze. Questi elementi aiuteranno in futuro, nel momento in cui si hanno dei buchi neri su alcune funzionalità Non trasferite. Valuto che secondo me il rapporto umano, quindi L'affiancamento sia comunque alla base di tutto. Quello che si deve riuscire a ottenere A fianco dell'AFFIANCAMENTO scusando, il gioco di parole. È una documentazione snelle ma completa Che possa aiutare laddove C'è stata una mancanza di di informazione trasferita. Ripeto, non è il 100% della soluzione e non risolve il problema. Ci sarà sempre comunque una perdita di informazione perché il telefono senza fili rimane un gioco valido per dimostrare che le informazioni vengono perse nei vari Step, anche perché ognuno interpreta e ha un suo modo di analizzare e Interpretare e avere una definizione di quello che ha appreso Quindi, ovviamente, quando io trasferisco la trasferisco a un'altra persona in parte come io lo ho capito, come io l'ho appreso e dall'altra parte si ha ovviamente un flusso interpretativo di quello che uno va a dire di quello che uno trasferisce e questa interpretazione porta a una modifica dell'informazione immagazzinata dall'altra persona e quindi potenzialmente Una modifica dell'informazioni o comunque. A un'informazione non dico degradata, ma un'informazione diversa dall'elemento che ne ha creato la dall'elemento sorgente dell'informazione.

R

Ok, d'accordo e durante il progetto sono C'è stato un grande turnover di risorse DEV. Secondo te questo ha portato delle grandi inefficienze? Tu hai trovato difficoltoso Adattarti a questa nuova risorsa, la risorsa, magari adattarsi a te. Insomma, come l'hai, come hai trovato questo scambio di di risorse?

I

Da una parte TI direi che Avere a che fare interfacciarsi Con più persone, con persone diverse Può essere una cosa positiva perché vai ad accrescere le tue conoscenze, perché ovviamente stando a contatto con un'altra persona, io insegno, ma imparo anche e quindi c'è uno scambio, quindi partiamo dalla parte positiva Del turnover della parte degli sviluppatori, perché ovviamente, andando a leggere poi il loro codice, verificando le le funzionalità Vai a apprendere modi diversi di approccio al problema, quindi, proprio banalmente, il problem solving che può essere diverso. Io il mio modo di approccio al problema, un'altro sviluppatore ne ha un'altro e quindi posso accrescere le mie competenze anche nel Diventare più bravo, più veloce a valutare e verificare l'attendibilità del codice sviluppato da altre persone. Quindi, avendo più persone, ovviamente si ha anche più skill diverse. È possibile che ci siano skill diverse, quindi apprendere anche o avere la possibilità di assegnare dei task a una persona piuttosto che un'altra, perché ha delle capacità appunto

diverse e magari ovviamente se si ha anche necessità di stare in tempi stretti, avendo più persone e competenze diverse Può aiutare a diversificare Come assegnare i task e come andare a Ad assegnare appunto i vari sviluppi. L'evento negativo però, avendo tante persone si hanno tanti incastri, tante sovrapposizioni e quindi gestire più sovrapposizioni ovviamente non è facile, quindi crea un Collo di bottiglia anche da parte mia nella gestione delle attività e nelle interconnessioni, nelle interdipendenze e quindi un po' Lega il continuo di uno sviluppo ad una persona. Perché questo re lo dico perché è la parte negativa, ossia. Parlavamo prima di trasferimento di conoscenze, se una persona ha sviluppato una sezione dell'applicativo, ovviamente ha un background, ha una storia Di quella funzionalità di quella sezione Al 100%. Perché ha iniziato a sviluppare lui ha iniziato a vedere il functional design, ne ha fatto analisi, ha riscontrato eventuali criticità Nel disegno funzionale e quindi È andato a risolvere tecnicamente Magari alcune situazioni che non erano ben chiare Nel documento funzionale e di nuovo se il documento funzionale è già stato fatto andare avanti fatto approvare al cliente, quindi magari Si hanno delle altre mail, si hanno degli altri flussi di comunicazione che vanno a sistemare quella quella mancanza di informazione, di analisi all'interno del documento Che rimangono racchiuse in due punti: nel codice E nella persona che lo ha sviluppato. Quindi ora immaginiamoci che questa persona non lavori più sul progetto, deve fa cambiare progetto, Tutta quella sezione che è stata fino adesso sviluppata da un soggetto deve cambiare ownership, deve cambiare persona E quindi sviluppatore. Cosa succede? Lo sviluppatore nuovo deve avere. Uno scambio di informazioni con lo sviluppatore precedente e come abbiamo detto prima, questo non è efficace al 100%, quindi si ha una perdita di informazione. A quel punto Lo sviluppatore ha a disposizione il codice. Ma molti sviluppatori fanno fatica a leggere il codice degli altri e averne una chiara interpretazione a livello Semantico, appunto, di cosa voglia dire della funzionalità di quello che fa. Quindi, richiede dello sforzo da parte della persona Di interpretare il codice scritto Da un'altra da un'altro soggetto, per poi poter applicarne delle modifiche. Quindi ovviamente c'è un'inefficienza. Quindi Il lato positivo Conosci persone, Impari da più persone, c'è più scambio. Lato negativo, si ha un'inefficienza nello sviluppo, sia ha delle informazioni che devono essere reiterate Perché c'è un nuovo soggetto che non ha tutte le competenze precedenti E quindi un'efficienza a livello di tempistica, perché diventa anche complicato lato mio dove Inserire un'altro soggetto su quella funzionalità, cercando di informarlo di tutto quello che serve, quindi ancora peggio, Fammi dire, se il soggetto che viene inserito all'interno del progetto è un soggetto che non ha mai visto quella tipologia di progetto, non ha mai lavorato in quel progetto. Perché, come dicevamo prima, una bassa conoscenza o una overview troppo bassa, troppo superficiale della funzionalità del progetto porta a non avere da parte dello sviluppatore un'analisi critica e quindi identificare possibili ripercussioni delle proprie attività, delle proprie funzionalità sviluppate con altre funzionalità

<p>dell'Applicativo. È complicato andare a gestire Questo principalmente per l'inefficienza che può portare allo sviluppo stesso del progetto.</p>	
<p>R</p> <p>Ok, molto chiaro. Parlando con uno dei dev, è uscito fuori che Soprattutto nelle parti nuove del progetto, come ad esempio la warranty, dove non c'è bisogno di avere un grande background sul progetto per fare delle scelte tecniche di più alto livello. Tu gli hai dato più possibilità, hai delegato a lui la responsabilità di queste scelte Proprio perché non dovevi trasferire delle informazioni nei suoi confronti. Il collegamento che mi viene di fare è che nel momento in cui Si trova il modo di effettuare un knowledge transfer verso una risorsa che conosce l'intero applicativo, Tu hai la possibilità di delegare verso di lui molte attività e Toglierti dall'onere a volte di essere un po il collo di bottiglia, perché ovviamente non puoi fare 1000 cose quindi mi chiedevo se tu trovi funzionale la possibilità di delegare dell'attività di scelta tecnica ai developers quando effettivamente hanno abbastanza conoscenze oppure è sempre un rischio, è sempre qualcosa su cui andare cauti.</p> <p>I</p> <p>Come dicevamo, l'applicativo ha diverse sezioni. Il core del nostro applicativo, quindi il nostro CPQ Ha delle funzionalità molto molto correlate l'una all'altra. Ma le funzionalità che sono state aggiunte Fuori dal CPQ che appunto parliamo di un portale per i clienti Sono ovviamente slegate da questa principale. Quindi Esistono sezioni dell'applicativo che sono tra virgolette, stand-alone che possono vivere di vita propria E che quindi possono essere separate e sviluppate in un certo modo Ma è vero anche che il core dell'applicativo ha delle funzionalità base che sono condivise. Quindi esiste La possibilità di andare a delegare una parte di questi sviluppi, ma Se gli si ragiona isolando completamente lo sviluppo di questo. Si potrebbe andare ad avere una situazione in cui si rimplementa una funzionalità già esistente. Questo ha dei suoi vantaggi e i suoi svantaggi Perché ovviamente Il connettersi con delle funzionalità esistenti porta a doversi interfacciare a doverle conoscere e a condividere il bug. Tenerle separate, ovviamente Isola queste funzionalità e ognuna vive di vita propria. Il bug dell'uno non non interferisce col bug dell'altro. Ma ne aumenta i costi e i tempi di sviluppo. Quindi Sì, avere margine di azione, avere Una sezione dell'applicativo che possa essere ragionato e sviluppata da una persona Può andar bene, Ma quella persona deve comunque esporre A me un'analisi Di implementazione che vuole andare a fare. Questo perché Io posso andare a recuperare alcune criticità, non evidenziate dalla sua analisi, posso migliorare alcune integrazioni con alcune funzionalità esistenti dell'applicativo Ed evitare che queste vengano implementate più volte. Quindi il problema principale di una conoscenza parziale dell'applicativo del progetto è che è possibile andare a trovare una situazione tecnica per risolvere una funzionalità. Ma questo potrebbe implicare un</p>	<p>TECHINCAL LEADER COLLO DI BOTTIGLIA</p> <p>Le parti nuove del CPQ non necessitano di grande informazione pregressa, quindi il lavoro può essere facilmente delegato</p> <p>Delegare il lavoro potrebbe portare comunque ad inefficienze, in quanto si potrebbero re-implementare funzionalità già sviluppate</p> <p>Delegare aiuta le risorse a crescere</p> <p>Affiancamento TL ha snellito il lavoro ma ha ridotto efficienza perché non conosceva tutto l'applicativo</p> <p>Per ridurre carico, rilasciare senza Code Review, caricando maggiormente i test. Questo riduce qualità applicativo</p>

Costo maggiore, un'analisi diversa perché Il costo tra virgolette, la parte di Stima dell'attività viene prodotta da me, quindi tra virgolette, io ho un'idea Di base di limiti che mi pongo e stimo, un'attività anche tenendo conto di alcune interconnessioni o di funzionalità già esistenti nell'applicativo. Dare completamente libertà a una persona di implementare una sezione può andar bene solo quando non ci sono completamente Possibilità di interconnessioni con il sistema. E comunque Facendolo così, ti dico che ho riscontrato delle problematiche. Per esempio, sulla parte di sviluppo dei contenuti provenienti dal CMS ovviamente è stato tutto sviluppato in maniera a sé stante Senza tener conto dell'applicativo, ma poi è servito identificare L'appartenenza di un contenuto a un certo mercato e quindi la parte di filtrare questi contenuti per questo mercato, che è un un'informazione, un elemento che non è presente all'interno dei contenuti e quindi esiste un'interconnessione. Quindi vero, il l'importare contenuti il visualizzare contenuti tutta questa robina qua è interessante, Figo può essere standalone. Ma non Al 100%, ad esempio la localizzazione Il codice lingua da utilizzare E qui si va a un concetto di Disaccoppiamento che deve essere preso in considerazione il codice lingua non è quello dell'applicativo, quindi l'applicativo di per sé, gestisce la lingua en-us, ma la lingua che viene utilizzata come codice identificativo per i contenuti del CMS è diversa e quindi devi sempre codificare, transcodificare questa informazione. Dove metti questa informazione? La metti a Monte, quindi chiedi ai contenuti con la lingua corretta, in modo da sapere che it per me è IT dall'altra parte è ita o quest'informazione la metti poi sul servizio che è stato creato, ma quindi non crei disaccoppiamento, perché quel servizio dovrebbe conoscere solo le informazioni che lo riguardano, Non dovrebbe sapere che un applicativo chiama l'it ita oViceversa, quindi ci sono degli elementi di Orchestrazione, che devono essere presi, messi in un punto centralizzati e gestiti Che è l'unico che deve effettivamente poi gestire Questo orchestrare i vari componenti che parlano l'altro e poi ognuno si può essere sviluppato con le proprie peculiarità, le e funzionalità del caso. Ma ti ho tirato fuori un'altro elemento che potrebbe essere completamente slegato dall'applicativo ma ha dei punti di contatto come la visibilità dei contenuti per mercato e il codice della lingua da visualizzare per quei contenuti Che è un'informazione Che non è completamente a sé stante. Devo avere uno scambio di informazioni. Quindi devo trovare un modo di disaccoppiare queste informazioni. Trovare un codice, un elemento, qualcosa che possa disaccoppiarlo e utilizzare per rendere lo sviluppo di queste funzionalità non interdipendente ma semplicemente connesso l'un l'altro tramite un disaccoppiamento. Se non hai tutte le informazioni, come funziona, che cos'è un mercato, diventa complicato andare a delegare il 100% di questo sviluppo a un'altra persona, di nuovo se non conosce al 100% del prodotto e comunque Un'analisi fatta da uno sviluppatore deve essere comunque secondo me validata da un'altra figura che lo posso fare crescere la prima volta delle analisi magari non andrà bene, la seconda neanche, da qui in poi si avrà una crescita dello

sviluppatore per cui le conoscenze sull'applicativo, le sul progetto sono aumentate e può aumentare anche la Affidabilità dell'analisi. Però è un processo che deve crescere.

R

quindi Ci sono stati poi dei momenti Durante il progetto in cui Sei risultato il collo di bottiglia perché avevi troppe cose da fare e c'è qualcosa che tu hai potuto fare per snellire il tuo carico o che l'azienda si è adoperata a fare per snellire il tuo lavoro?

I

Ho accettato quello di andare a aggiungere una figura simile alla mia che andasse a seguire parte dello sviluppo. Questo di nuovo, anche se una figura come la mia, che ha la mia stessa seniority, le mie stesse capacità tecniche e skill di nuovo non conosce il progetto al 100% come lo posso conoscere io. Questo comunque ha portato dell'inefficienza a posteriori. Però un'iniziale Metodo un iniziale approccio è stato quello di andare a splittare la funzionalità Dell'hero In più Hero che possono andare a seguire più sviluppatori, in modo da avere quello che dicevamo all'inizio, cioè la mia figura che deve comunque supportare lo stallo, supportare la parte di sviluppo e quindi aiutare nel momento del bisogno e di analisi o comunque nella parte tecnica e di crescita anche delle persone all'interno del progetto, ma anche all'interno dell'azienda, le persone devono crescere, Per andare a snellire il carico in quanto c'erano un buon numero di sviluppatori sul progetto. Quello che poi è stato fatto, in realtà che secondo me è stata una perdita di qualità, Quindi non è stato fammi dire Una liberarmi dal carico ma di snellire la forza di lavoro che applico Nella code review, per cui abbiamo fatto rientrare [sono stati rilasciati] una serie di sviluppi Anche molto grossi, quindi, comunque, che avrebbero richiesto grosso periodo di code review in quanto appunto c'era molta parte di codice che era stata toccata e quindi aver fatto rientrare tutta una serie di funzionalità Per essere appunto testate dalla parte funzionale. Questo mi ha Snellito la mia funzione d Code Review. Però anche questo diventa complicato, perché poi se introduci una buona mole di funzionalità all'interno del dell'applicativo, poi queste devono essere testate, quindi comunque. Collo di bottiglia? No, ma lo diventa qualcun altro comunque. Verso il lato funzionale rispetto che il mio lato.

R

Eh però, a livello di qualità del codice, questo non porta comunque dei problemi?

I

Sì, banalmente nelle performance quindi devi fare 5 cicli, ne posso fare uno invece di fare 10 queries ne posso fare una e quindi questo a livello tecnico il mio compito è anche quello di valutare. La possibile implementazione, l'implementazione fatta e magari cambiare l'implementazione fatta perché appunto Non rispetta dei

criteri di performance, che magari potrebbero essere migliorati. Quindi sì se non faccio tutte le mansioni per cui Sono l'owner all'interno del progetto, Ovviamente ci può essere una mancanza, cose che poi vengono riscontrate dopo, perché magari su funzionalità successive o su elementi successive vado a rivedere quello che è stato sviluppato e magari poi vado a metterci una pezza. Però problemi della tempistica, poi diventa una scelta, quindi Non ci sono molte strade, cioè se un'unica persona che fa un lavoro, sei un collo di bottiglia sì, e quindi l'unico modo per andare a migliorare questo processo è andare a snellire ancora di più le funzionalità da sviluppare. Spacchettare le funzionalità in Task più piccoli, in modo che questi possano avere un continuo rientro all'interno della del flusso principale di sviluppo, Quindi che si possono testare piccole porzioni dell'applicativo invece di andare a modificare l'universo mondo e poi a dover andare a testare tutto assieme. Il problema di questo è che ovviamente, se crei delle release, delle wave, un processo di sviluppo, devi anche rendere stabile quello che stai rilasciando e quindi se vai a introdurre tante modifiche all'interno del progetto, questo potrebbe creare delle regression, potrebbe creare delle funzionalità disattese. O piccoli bug, piccoli glitch, piccole modifiche visuali che magari devono andare a essere sviluppate in wave future del progetto e quindi questo crea poi un problema Nell'esposizione al cliente dell'Applicativo per cui ci sono cose a metà, elementi non funzionanti e quant'altro, quindi da una parte sarebbe bellissimo avere piccole porzioni di codice da verificare, controllare e far rientrare nelle piccole funzionalità, lo spacchettamento di funzionalità molto complesse in funzionalità più piccole per poterle controllare e verificare e testare. Dall'altra, se ho N sviluppi in parallelo per N funzionalità, questo diventa complicato. E anche lo spezzettare la funzionalità in più elementi che magari non sono self contained diventa complicato perché poi in generale, per andare a vedere quando è stata introdotta quella funzionalità si ha avuto questo bug, quindi bisogna trovare un compromesso. Però Ci sono delle attività che richiedono del tempo. E quindi Se sei un collo di bottiglia, sei un collo di bottiglia perché la il tuo compito è quello. Non saprei dirti se ci fossero N Hero che fanno quello, Ti direi l'unico modo di nuovo sarebbe che le funzionalità che ogni Hero vede sono self-contained Quindi non c'è effettivamente nessun interconnessione, se si ha la possibilità di non avere nessuna interconnessioni Tra l'attività di sviluppo, allora sì, quello buono si può fare appena. Appena viene introdotta una zona grigia perché l'ownership è di più persone in questo caso si ha una situazione di un sistema complicata perché l'ownership è condivisa e diventa complicata andare a spezzettare e splittare, Dividere le attività, quindi poi testare. Non saprei dirti se c'è un modo fisico vero, dovuto alle esigenze di progetto anche di andare a migliorare il flusso lavorativo. Per non essere un collo di bottiglia, perché di nuovo andassi anche a fare la Code review ogni giorno di quello che fanno le persone ovviamente oggi fanno una cosa, domani ne farò un'altra, domani un'altra ancora e io ho tutto il codice che hanno

<p>cambiato dall'inizio alla fine. È quello che hanno cambiato il giorno dopo, magari hanno toccato lo stesso file. Quindi io non avrei comunque la capacità di valutare in maniera intera la funzionalità, quindi verificare se è attendibile o meno. Potrei verificare il pezzettino singolo, ma non avendo l'overview generale, cioè non saprei dirti se effettivamente quello che è stato sviluppato è attendibile. Se si riuscisse a chiudere le funzionalità, quindi avere dei pacchetti stagni in cui si sviluppa, si testa si avanti così, quello potrebbe essere l'elemento risolutivo. Alla attività mia, ma anche all'attività dei paladini, per poter essere sicuri che quello che stiamo andando a fare funziona col grande interrogativo che alla fine che tutti questi pezzettini la funzionalità al 100% deve funzionare. Per cui è vero che puoi testare i singoli pezzettini singoli, pezzettini, funzionano come ti aspetti, ma non è detto che poi la funzionalità finale funzioni. Come ti aspetti. Perché magari effettivamente tu ti aspettavi che uno di questi pezzettini funzionasse in un certo modo, Ma in realtà hai pensato un modo sbagliato in cui funziona e non si collega nella catena. Il problema sono le interconnessioni che si hanno nello sviluppo. Ho importato un dato, sono sicuro che va bene. Il dato è importato, c'è. Alla fine del processo mi accorgo che vero che il dato è stato importato, ma nella visualizzazione io devo raggruppare un tot di cose per un certo modo e mi sono perso questa informazione, quindi tra virgolette l'import lo dovrei cambiare perché in quel modo riesco a migliorare, a velocizzare il processo oppure no, A livello visualizzavo sto facendo vedere delle robe diverse, modifico la query che va a selezionare questi dati, ma la modifica di quella sezione qui mi crea una situazione di visibilità che non avevo pensato e quindi più elementi o di interconnessione più questo diventa difficile splittare, gestirlo e portarlo avanti</p>	
<p>R Abbiamo parlato delle interazioni che hai con ogni membro del team, ma non con i jedi. Trovi che il processo di passaggio di informazioni di quelli che sono i vincoli tecnici da lato tuo al lato suo, ma anche viceversa. Quindi c'è il jedi pensa che sia fondamentale nel processo della customer experience. Pensi sia un processo che funzioni bene, ci sono delle inefficienze, Quali sono?</p> <p>I La verità è che la parte dei jedi ha un impatto iniziale. Alla partenza delle attività, quindi, già nell'analisi della funzionalità e quindi nella stesura del FD e nella decisione del funzionamento dell'applicativo vengono introdotti i jedi e quindi loro portano la loro idea di funzionamento di UX. E lì lo scontro titanico è come rendere la parte tecnica a supporto della parte jedi di flusso disegnato dai jedi e viceversa. Lo scontro è titanico, perché a volte alcune funzionalità come vengono disegnate richiedono un effort di sviluppo troppo elevato. Quindi, Eh, si deve scendere a compromessi. Per cui. Normalmente l'Hero cerca dei compromessi con la parte paladini. Perché il requisito funzionale ovviamente verso il cliente deve essere chiaro, ci sono dei limiti tecnici, c'è un costo. E poi,</p>	<p>INTERAZIONE CON I JEDI</p> <p>Esiste uno scontro tra il modo in cui la funzionalità può essere implementata dall'Hero e il flusso desiderato dallo Jedi, che va risolto in un tavolo a tre.</p> <p>Dopo la fase di gara, risulta molto complicato trovare degli accordi tra le tre parti del progetto, in quanto hanno background e visioni differenti</p>

ovviamente, dopo che si ha questa raccolta di informazioni, si passa anche dai jedi che vanno a identificare una possibile soluzione a requirement del cliente. La richiesta del cliente. E questo poi ti crea uno scontro perché appunto la parte tecnica a parte degli hero hanno delle difficoltà a raggiungere quella tipologia di funzionalità disegnata come la chiede lo Jedi, perché a livello tecnico sarebbe troppo costoso quindi lì effettivamente c'è un trio Che dovrebbe cercare di lavorare il più in sintonia possibile, Fammi dire, Per poter Capire e mettersi nei panni delle altre figure In modo da trovare tutti insieme un accordo e una soluzione al problema. Quindi sì, c'è un'interazione da parte mia con il dipartimento Jedi E sì, ci sono degli degli scontri, Dei flussi che si si creano tra paladin, Jade, Hero per Trovare la soluzione che possa avere un impatto massimo verso il cliente, una user experience al top e anche un costo contenuto, Perché il problema di tutto questo è che Se io non avessi dei limiti di budget E dei limiti di tempo di sviluppo potrei anche dire va bene, disegnatelo quello che volete, lo andiamo a implementare. Ma siccome deve esserci un buon un buon trade off, un buon compromesso tra quello che è il costo E quello che è la funzionalità è quello che è la user experience, le tre figure devono lavorare per ottenerlo.

R

C'è stato un momento in cui durante il progetto Questa comunicazione è andata male? Il trade off non è stato fatto correttamente?

I

All'inizio del progetto Ma perché Ovviamente, comunque parliamo di una gara Quindi Prima di avere una commessa, ovviamente devi partecipare a una gara. E Il flusso Di gara ovviamente è diverso dal flusso di progetto Gli step che Si sono verificati appunto nella fase di gara, si sono configurati nel Sviluppare una serie di visual, funzionalità, Elementi, sezioni facente parte del del progetto, con dei criteri, con, con con un'analisi. Più dal punto di vista, appunto che Possa essere la la user experience o comunque Un applicativo disponibile appunto per il customer e quindi Tutta una serie di funzionalità, dashboard, cruscotti Attività in generale che tengono traccia, tutta una serie di elementi Che dovevano far parte del progetto. Quindi in quella fase lì manca la parte tecnica per cui Si va a presentare una tipologia di prodotto. Questo poi ha portato a successive iterazioni quando è stata vinta la gara e quindi poi si parlava di progetto Sviluppando delle funzionalità, sviluppando dei visual, che tenesse conto invece dei vincoli tecnici. Quindi lì Si è iniziato a creare Un po' più l'interconnessione. E comunque, forse in quella fase si è mantenuta un certo distacco dalla parte tecnica alla parte di Design Perché ovviamente è stato chiesto rispetto alla porta di design, di produrre una stima. Che però poi si va a Avere un costo troppo elevato, come dicevo prima, per la parte di design e quindi si è richiesto un compromesso e quindi poi si è Ripensato a quello che si poteva fare e ridisegnati visual. Quindi ci sono state

<p>varie interazioni che hanno portato alla modifica del flusso modifica dei visual. E secondo me in questo processo qua, non non so come, bisognerebbe migliorare L'interconnessione che si ha tra tutti i membri del team Per garantire questo trade off in un tempo minore fammi dire, quindi con meno iterazioni</p> <p>Per cui Si è provato disaccoppiando le varie cose si è ragionato Da un punto di vista funzionale, di design e poi da un punto di vista tecnico si è, si è andato introdurre questo. è difficile far passare alla parte di design i vincoli tecnici e è difficile far passare nell'ambito funzionale anche i vincoli tecnici, perché a volte Non si ha la percezione dell'effort richiesto per fare certe cose, o a volte basterebbe soltanto trovare un Più flessibilità mentale, quindi una piccola differenza in quello che si è disegnati per andare a diminuire il costo, ma mantenendo Un'esperienza al top. Ed è quello che dicevo all'inizio, cioè è difficile mettersi nei panni degli altri elementi che fanno parte di questo tavolo Per avere una collaborazione efficiente. Perché alla fine l'efficacia si raggiunge nel senso che il risultato lo si ottiene con qualche compromesso, ovviamente. Però questo dovrebbe essere più efficiente, nel senso che bisognerebbe migliorare Il tempo per arrivare alla soluzione Ci sono stati dei compromessi, si è fatto questo lavoro. E in generale, quando poi io vado a interfacciarmi con la parte di design Ovviamente do le mie indicazioni e i vincoli tecnici. Quello che diventa difficile è reiterare Il processo in modo trovare una soluzione che soddisfi non al 100% ma che soddisfi l'utente E la parte di reiterazione è quella che occupa più tempo e quella meno Efficiente in questo processo</p>	
<p>R Molto chiaro, molto chiaro Volevo poi chiederti se Credi che il team abbia imparato da se stesso durante il progetto? Sono stati anche utilizzati degli strumenti per affinché il team potesse riflettere su se stesso, come possono essere i 1-to-1 o anche le retrospective, ma credi che abbiano funzionato? O sono stati superflui?</p> <p>I Il Team è stato poco fisarmonica, quindi si è espanso, Si è contratto quindi non avendo una linearità sul team è un po' difficile rispondere a questa domanda perché non avendo lo stesso team fino alla fine ci sono un sacco di variabili Nell'intermezzo Che rendono difficile rispondere alla domanda, quindi siamo migliorati perché abbiamo utilizzato questi strumenti? È migliorata la coscienza? Ci sono degli elementi che sono rimasti all'interno del dello sviluppo Per la durata del progetto, però, sono cambiate un po le interconnessioni</p> <p>R Cosa intendi?</p> <p>I Sono cambiati gli Hero, sono cambiati i Paladin Ci sono state tante interconnessioni, per cui è Cambiate molte volte La struttura di</p>	<p>TEAM IMPARA DA SE' STESSO</p> <p>Retrospective e 1-to-1 hanno sicuramente aiutato i singoli membri del team a crescere</p> <p>Difficile dire quanto sia migliorato il team, visto che è cambiato molto nel tempo</p>

<p>progetto del team per poter dire che la struttura di progetto del team è cresciuta e ha imparato da sè stesso. Posso dire che Gli strumenti 1-to-1 e retrospective ci hanno aiutato a far emergere le problematiche che erano presenti nel progetto nel team. E in parte hanno anche aiutato a Venirne a capo Su alcune criticità più comuni che affliggevano Di più le persone. Non so dirti se poi alla fine abbiamo raggiunto il risultato perché il team è cambiato, quindi non si ha una stabilità tale per cui posso dirti che abbiamo migliorato il 100% di quello che abbiamo trovato nei vari 1-to-1 e retrospective.</p>	
<p>R</p> <p>Guardando il budget di progetto questo ha iniziato a salire nei periodi tipo verso Marzo, ad agosto ha toccato lo zero, quindi erogato e venduto erano uguali. Per poi scendere. Andare sotto lo zero. Quali pensi possano essere motivazioni per il quale si è erogato più di quello che si è venduto.</p> <p>I</p> <p>Diverse. Sottostimato. Un'analisi Parziale. Che quindi ha portato ad avere delle complicazioni nel mentre, Cambi in corso; quindi, richieste Funzionali che possono aver variato un po lo scopo Un turnover Dei vari elementi che fanno hanno fatto parte del team Costi tecnici per l'utilizzo di tecnologie più nuove e l'integrazione di queste, che ha richiesto una crescita degli elementi all'interno del team appunto, non tutti con le stesse skill, quindi Richiesta di conoscenza dovuto a integrazione con elementi nuovi. Inefficienza dovuta appunto a interconnessioni. Come dicevo prima, tra i vari sviluppi e quindi il l'incastro delle varie funzionalità. Inefficienza in generale dovuta magari a comunicazione a pianificazione della delivery delle varie attività. Nel senso che. Dato una stima, dato un budget. Avendo un'analisi Un po ad alto livello, perché ovviamente è un progetto enorme Porta della difficoltà nel calare Quella stima quell'attività. Su una persona che possa avere il 100% di conoscenze per essere efficace al 100% E anche Un bagaglio tecnico al 100%. Perché possa essere efficace su quell'attività. E sempre a livello di pianificazione, Come dicevo prima, una gestione Più a stretto giro della funzionalità sviluppata e rilasciata in modo da Identificare eventuali criticità di quelle attività prima Che Vengono effettivamente esposte Per cui bisognava andare a Chiudere molto di più l'analisi e spendere più tempo nella parte di analisi, quindi magari sottostimata Per poter appunto dare degli FD e uno scopo molto chiuso E dare uno scopo molto chiuso anche allo sviluppatore che possa portare il risultato.</p> <p>R</p> <p>Quindi cosa pensi che mancato nella parte di analisi? Cioè cosa fare affinché si riesca ad analizzare bene il requisito</p> <p>I</p> <p>Avere un'idea di requisito molto molto chiara Per raggiungere uno scopo. Il progetto è nato con un requisito Molto lasco. E molte cose</p>	<p>BUDGET</p> <p>Bilancio negativo perché progetto sottostimato</p> <p>Difficile effettuare stime precise su analisi ad alto livello</p> <p>Requisiti iniziali del progetto molto larghi</p>

sono state poi Risistemate, riviste nell'arco della stima, Nella proposta di stima, la proposta di di progetto. Tra virgolette, andando a Sviluppare un requisito Dovuto all'esigenza del cliente, Ma non con un requisito da parte del cliente. Quindi anche lì da parte del cliente... nessun cliente sa cosa vuole, però Uno scopo molto chiuso su quelli che devono essere le funzionalità Richieste e un supporto da parte del cliente di come queste funzionalità nella loro idea di business Andassero a impattare. Questo ha portato ad esempio molti, molti rework, molte modifiche, per cui noi abbiamo fatto una proposta di di soluzione Che poi Dopo essere implementata, Ha portato magari a qualche Richieste da parte del cliente. Quindi secondo me non esiste poi fisicamente, praticamente un modo Per avere il 100% di questo dato dal fatto che le persone cambiano idea. Quindi avere un'analisi forte, dovuta da requirement forte, se hai dei requirement forte da parte del cliente, il cliente sa cosa vuole, hai uno scopo molto chiuso. Lo scopo di progetto È stato più ampio di quello che poteva essere un requisito molto, molto chiuso. Questo porta tante variabili e quindi Una difficoltà Pratica nella andare a stimare un'attività In un certo modo. In più, avendo risorse che cambiano durante il tempo, questo crea ancora più più inefficienza. Ma non avessimo avuto una problematica di questo tipo Si sarebbe secondo me dovuto lavorare molto di più sull'analisi e comunque andare a stimare dopo aver fatto un'analisi completa L'attività. Perché solo Se hai un'analisi completa dell'attività e dell'interconnessione che puoi avere, puoi andare a tirar fuori una stima attendibile. Una stima data da delle assunzioni Seppur cercando di essere più strette possibili non completamente diciamo derivanti dal cliente Possono poi portare a Una deriva sull'impatto finale. Quindi Ci sono tante cose da mettere in conto, appunto inefficienza, l'apprendere, tecnologie nuove e cose di questo tipo.