

Lezione del 2024-11-04-20241104_185103- Registrazione della riunione

4 novembre 2024, 05:51PM

1 h 29 m 40 s

● **SANTE DOTTO** trascrizione avviata



SANTE DOTTO 0:14

Allora, come come vi anticipavo?

Oggi faremo una sessione di analisi di un approccio metodologico alla realizzazione dei progetti a cui viene dato questo nome, che leggete sulla slide introduttiva di Agile project management, quindi gestione dei progetti agil.

E nel fare questo faremo anche un parallelo con le.

Le diciamo metodologie, i processi più tradizionalmente utilizzati, in particolare nello sviluppo di sistemi informativi, proprio per sottolineare le differenze e anche le modalità di applicazione, quando è giusto applicare questa metodologia agile?
Angelica, Buonasera.



AG **Angelica Giroto** 1:05

Buonasera, mi scusi il ritardo?



SANTE DOTTO 1:07

E per carità, abbiamo appena iniziato guarda Eh, come vedrete la cosa un po sorprendente da un certo punto di vista, per altri aspetti molto molto, molto meno.

Perché poi sia la fase di progettazione che la fase di realizzazione dei progetti insistono su un contesto come quello della trasformazione digitale delle organizzazioni che si è andando ed è andato evolvendo proprio con lo sviluppo di tutta una serie di servizi in rete e la progressiva il progressivo utilizzo anche in mobilità, di applicazioni e di di sistemi informativi con delle tecnologie che fino a poco tempo prima erano, come vi dicevo, chiuse nei laboratori di sviluppo.

È ormai da diversi anni.

Cominciamo a utilizzarle sui nostri smartphone, quindi questa facilità di utilizzo che si è resa possibile di tecnologie più evolute ha portato anche delle modifiche profonde negli approcci alla progettazione dei sistemi informativi e del e alla realizzazione di questi.

Di questi progetti diciamo più innovativi rispetto al passato.

Sostanza si è un po' alzata l'asticella di utilizzo delle tecnologie digitali all'interno delle organizzazioni e nella società.

Questo, di fatto è avvenuto.

Tutto quello che prima era un'informatica legata più che altro all'automazione di procedure, chiamiamole amministrative, per capirci, all'interno delle organizzazioni è stato superato con un'evoluzione che ha portato a utilizzare l'informatica per tante altre funzionalità che fino a una decina ventina di anni fa non diciamo la verità, non erano neanche prevedibili. Ecco.

È un parallelo che faremo tra gli approcci legati alla progettazione, per esempio col design thinking e allo sviluppo dei progetti, per esempio con la già il project management.

Ci faranno ritrovare tutta una serie di competenze soft che sono alla base di questi, di questi nuovi approcci, sia nella progettazione che nella realizzazione delle iniziative più innovative.

Allora io condivido perché io ho detto, come vedete dalla Slide, ma nessuno mi ha detto che non l'avevo ancora condivisa, per cui eccola qua.

Adesso dovrete vederla già il project management.

Vedete, tutti sì.

Qualcuno che mi faccia un cenno con la testa che vedete la slide perfetto grazie vabbè.

 **AG** **Angelica Giroto** 3:48
Sì, sì.

 **GP** **Giovanni Pecoraro** 3:49
Ci vediamo.

 **SANTE DOTTO** 3:51
Qui, come al solito, vede un po' i riferimenti da cui ho tratto questi spunti EE cominciamo un po' a vedere in che contesto nasce questo nuovo approccio metodologico.
Nasce nel contesto dello sviluppo software.
Prima cominciamo col dire, quindi a guidato nella nella Eh.
Definizione di questa metodologia per la gestione dei progetti.

E lo sviluppo l'approccio diciamo tecnologico, legato allo sviluppo di applicazioni informatiche e questo sistema è nato nel 2001 anche come un po' indecente. King, vi ricordate a cavallo dell'anno 2000 all'Università di Stanford?

Questo è 1 1.

Approccio invece, che nasce da un team di esperti sviluppatori di software e nel 2001 si è riunito fisicamente e producendo anche un manifesto per lo sviluppo agile del software per risolvere tutta una serie di tematiche che si stavano cominciando a manifestare nello sviluppo delle applicazioni software proprio in virtù di quella forte innovazione veicolata dall'utilizzo di tecnologie fino all'ora.

Non così spinte non così utilizzate in diversi settori dell'organizzazione delle imprese. Quindi la prima applicazione è stata proprio legata di questa metodologia di sviluppo alla quella che si stava cominciando a definire come una vera e propria trasformazione digitale delle organizzazioni.

Dopodiché vedremo questo approccio metodologico si è, diciamo in qualche modo ampliato.

E di fatto RE le modalità agili di sviluppo dei progetti sono state applicate anche al di fuori del contesto dello stretto contesto, dello sviluppo del software.

Quindi noi vediamo, abbiamo visto due vestiamo vedendo insomma due approcci, uno legato alla progettazione che partiva dal design e convergeva verso i sistemi informativi.

Quest'altro, nelle fasi di realizzazione che parte dai sistemi informativi e va ad ampliarsi, alla gestione di un progetto di innovazione, indipendentemente dall'utilizzo delle tecnologie digitali, tanto da definire una serie di approcci metodologici.

Eh?

Etichettati come business, agility, agilità nella trasformazione organizzativa nella trasformazione del business e dei modelli di business delle imprese.

L'approccio metodologico per la realizzazione di questi progetti è sostanzialmente legato agli stessi, agli stessi mindset, alla stessa modalità di gestione, alle stesse fasi di realizzazione.

E.

Se qualcuno di voi ha avuto modo nel suo percorso di studi, di affrontare il tema della qualità totale.

È un tema che si è diffuso nel nell'impresa a cavallo degli anni 80 90, il modello di riferimento è stata la Toyota, la Toyota è stata la casa automobilistica che per prima

ha sviluppato sistemi, modelli di progettazione e di gestione dei progetti e finalizzati al miglioramento progressivo e continuo della qualità dei prodotti realizzati.

In particolare le loro auto, e indirizzando in modo più diretto, più specifico, più studiato e le esigenze dei clienti che i clienti delle auto Toyota in qualche modo esprimevano.

E anche coinvolgendo negli aspetti legati, per esempio alla minimizzazione degli sprechi, alla miglioramento dei processi di lavoro, a ridisegno delle logiche così ferree in un certo senso della produzione automobilistica legata alle catene di montaggio.

Quindi tutta una serie di riflessioni che coinvolgevano direttamente i dipendenti. Un approccio molto orientale, venivano creati anche dei gruppi di lavoro che, al di fuori dell'orario.

Canonico di di lavoro, riflettevano e proponevano al management miglioramenti nell'esecuzione delle varie fasi legate alla produzione di un'automobile o ai materiali utilizzati o alle modalità con cui venivano disegnati i nuovi modelli.

Insomma, era tutto lo sforzo che coinvolgeva nella Toyota di quegli anni e tuttora è tuttora.

Succede questa cosa EA tutti i livelli i dipendenti per ottimizzare i processi di lavoro e minimizzare gli sprechi, mettere.

Ciascun dipendente nella migliore condizione per esprimere le proprie competenze, quindi questi processi di qualità totale, come veniva definito in quegli anni, hanno poi in qualche modo ispirato in questo percorso dagli anni 80 a oggi.

E la nascita e lo sviluppo di metodologie di progettazione e di gestione dei progetti di realizzazione dei progetti come quelle che stiamo stiamo vedendo nell'ambito di questo corso, stiamo analizzando nell'ambito di questo corso.

Quindi questi 17 progettisti software guru dell'informatica si riuniscono nel mese di Aprile del 2001 in un albergo per provare a risolvere un problema comune.

Il problema sostanzialmente era come tenere conto nello sviluppo dei progetti innovativi più innovativi del ruolo centrale che in questa nuova economia che stava nascendo, l'economia digitale.

Prendevano le tecnologie digitali, appunto, e in particolare i nuovi fenomeni che si andavano sviluppando sulla rete.

Consentire alle imprese.

Rivedendo in modo importante processi di lavoro e sistemi informativi di sfruttare le potenzialità che la rete offriva.

Era l'obiettivo che questi, questo gruppo di sviluppatori voleva cercare di sostanziare, di descrivere, di analizzare e di realizzare con un approccio metodologico completamente diverso da quello seguito fino ad ora.

Per esempio, il tema era una gestione più diretta tramite la rete del contatto con il cliente finale.

È una puntualizzo anche all'interno dell'impresa, della potenzialità della rete, ad esempio, per delocalizzare alcune funzioni di lavoro.

Quindi persone si cominciavano in quegli anni a collaborare su progetti e condivisi nell'ambito dello stesso team di progetto, pur essendo dislocati su parte diverse dal punto di vista geografico.

Perché?

Perché si poteva cominciare a dialogare in rete, un po' come stiamo facendo noi oggi su teams.

La modalità di partenza, che in questo contesto di forte evoluzione creava problemi, era una modalità che veniva chiamata waterfall a cascata, in cui le fasi di sviluppo di un progetto di realizzazione di un progetto venivano eseguite una alla volta sequenzialmente e non era così facile.

Modificare in corso d'opera le specifiche da automatizzare per cui una volta partiti è come se si lanciasse un treno in corsa che si fermava soltanto alla stazione della del primo rilascio in esercizio del della della della fine di quel progetto, di quella di quella realizzazione, di quelle funzionalità che erano state richieste in un qualche misura, senza preoccuparsi di fare nessun check intermedio con gli utenti, con tutta una serie di di problemi legati ai tempi e ai costi di queste realizzazioni.

Quindi, man mano che l'asticella dell'utilizzo delle tecnologie digitali e all'interno di un'azione si alzava andando a coinvolgere funzionalità sempre più evolute.

E sempre più e un maggior numero di di utenti interni all'organizzazione crescente diventava necessario non procedere in questa modalità a cascata così rigida, sequenziale, ma potere interloquire diversamente con gli utenti e inserirli di fatto nel processo di sviluppo del software.

Vi ricordate qualche lezione?

Fa, dicevamo, in fondo, se ci pensate, tutte le grandi piattaforme di informatica che oggi rappresentano imperi economici importanti come Amazon, come Facebook eccetera, erano nate per esigenze individuando esigenze molto puntuali molto specifiche.

Amazon era nato per vendere libri Facebook era nato per trovare i compagni di

scuola, ritrovare i compagni di scuola.

E dopodiché si sono arricchite di funzionalità.

Queste piattaforme, che sono state rilasciate in esercizio, dicevamo dei beta test, come si dice in gergo, e gli utenti stessi attraverso l'utilizzo di queste piattaforme segnalavano problemi o funzionalità di cui avrebbero voluto fruire.

Quindi è una modalità di sviluppo molto diversa, molto ricorsiva.

Si, si ottengono dei feedback in un meccanismo di questo tipo che consentono in corso d'opera di modificare e migliorare e aggiungere o togliere alcune funzionalità inizialmente sviluppate.

Quindi i due processi li vediamo adesso un pochino più nel dettaglio, ma sostanzialmente il waterfall è un processo a cascata di fasi che si susseguono in modo lineare senza prevedere dei ricicli.

E la modalità già il nasce in qualche modo per rendere più facili e utili e gli indispensabili ricicli che queste nuove tecnologie comportavano perché gli utenti potevano richiedere molto più semplicemente nuove funzionalità, che tra l'altro dovevano essere rilasciate anche molto velocemente.

Perché l'impresa la piattaforma mantenesse il vantaggio competitivo rispetto ai concorrenti.

E questo il motivo per cui Microsoft, Facebook rilasciano continui aggiornamenti per essere sempre sulla frontiera dell'innovazione.

Sempre Google, sempre primi a rilasciare nuovi servizi e quindi bisogna velocemente recepire le indicazioni degli utenti che di fatto testano utilizzando i sistemi e sviluppare le nuove componenti e modificare RE e modificare le applicazioni e correggere gli errori che emergono velocemente.

Allora l'approccio tradizionale e l'approccio in cascata, qui vedete in questa figura ho elencato le fasi di progetto che partono dall'analisi dei requisiti utente fino alla manutenzione dell'applicazione, quindi sostanzialmente si parte raccogliendo i requisiti.

Molto spesso nell'informatica degli anni 80 e degli anni 90, e c'è stata questa stortura di cui vi ho parlato tante volte, ma estremizzo anche un po il discorso, ma per farvi rendere conto della profonda differenza rispetto alle metodologie più innovative.

La struttura era dovuto dal fatto che i detentori delle competenze informatiche in qualche modo questi requisiti, che pure gli chiedevano agli utenti ma di fatto li definivano.

Secondo la loro visione dell'utilizzo delle tecnologie, quasi cercando di anticipare, per usare un verbo, diciamo neutro o di forzare per usare un verbo un po' meno neutro. L'utilizzo di queste tecnologie, tecnologie digitali da parte degli utenti, quindi era un'analisi dei requisiti che non sfociava in una progettazione funzionale.

Stile design thinking, creativa profonda, riciclando.

Comprendendo, no era, era spesso un'analisi di requisiti, diciamo piuttosto sommaria. Dopodiché si partiva stendendo le specifiche tecniche, progettando qui la progettazione, intesa come progettazione tecnica, quindi di architettura informatica. Lo sviluppo del codice l'installazione di questi moduli, eseguibili sui computer che avevano gli utenti con delle tecniche che si andavano affermando, per esempio di distribuzione del software in rete.

Adesso noi non ci facciamo neanche caso.

Il nostro cellulare si aggiorna senza neanche chiederci nulla, anzi perdiamo il controllo del nostro cellulare perché questa cosa?

Personalmente, avendo vissuto l'informatica di diversi anni fa, mi infastidisce che il cellulare si aggiorni a mia insaputa, magari andando a modificare anche delle personalizzazioni che avevamo fatto, perché questo succede comunemente in quegli anni.

Questa modalità di distribuzione software e delle dei nuovi programmi, delle modifiche eccetera.

E si stava diffondendo anch'essa in rete e quindi non si viaggiava più con persone che prendevano aerei per andare a portare dischetti o dischi.

Proprio hardware con il software?

No, questa cosa si faceva Eh.

Si cominciava a fare in rete, anche questo è un cambiamento importante perché tutta questa tecnologia sul cellulare non potremmo avercela se non fossero diventati assolutamente sicuri ed efficienti i meccanismi di software Distribution e poi si passava alla manutenzione, quindi nel momento in cui si nello sviluppo terminava con un collaudo, si installavano le versioni definitive sui computer degli utenti e si procedeva con la manutenzione di queste applicazioni, che perlopiù poteva essere una manutenzione correttiva.

Perché gli utenti trovavano degli errori o una manutenzione evolutiva se gli utenti chiedevano nuove funzioni, però, di solito il rilascio di nuove funzioni era subordinato al completamento del ciclo di sviluppo.

Quindi, una volta sviluppata l'applicazione si lavorava nei dovuti tempi e nei dovuti

modi e diciamo compatibilmente con le esigenze della degli uffici.

Ti di information Technology di quegli anni si rilasciavano nuove funzionalità, quindi un processo anche molto lungo nel tempo per avere una nuova funzionalità.

Magari occorrerà aspettare mesi, se non anni.

Che cosa succedeva poi durante la vita di questo sistema informativo, sviluppato con questa modalità waterforce, queste sono delle statistiche che vi faccio vedere, l'ho raccolto da alcune sono sono, diciamo istogrammi, torte raccolte da società di consulenza di quegli anni che sostanzialmente dimostra una serie di di diciamo di disfunzioni.

Allora la prima è che di tutto il codice sviluppato soltanto il 45% in questa ricerca, ma diciamo meno della metà, ma forse anche la metà, capite bene che lo spreco era realmente utilizzata sempre.

Addirittura c'era un 19% in 7%, quindi un 26% che veniva utilizzato mai o raramente il 16% qualche volta il 13%.

Diciamo spesso, quindi si sviluppava 100 per utilizzare a malapena 50.

Questo da cosa dipendeva e dipendeva dal fatto che i tempi di sviluppo sono talmente lunghi con questo metodo, che magari le esigenze dell'utente cambiano, dipendeva dal fatto che le esigenze dell'utente non erano state ben comprese nel momento della analisi dei requisiti.

Come si chiamava la prima fase nella, nelle, nei, nei, nei processi di sviluppo vuote.

E è tipico nella fase di collaudo, era il verificarsi di una situazione in cui l'utente che partecipava al collaudo, a cui veniva mostrata l'applicazione sviluppata dopo X mesi dal momento della raccolta delle inquisiti, utente se non anni.

Affermava sostanzialmente qualcosa di molto simile, ma io non avevo capito che questa cosa avrebbe sarebbe funzionata così io avrei avrei voluto una cosa diversa per me questo processo doveva girare in una maniera diversa e lì succedeva il panico totale, perché i soldi erano stati spesi.

Le tecnologie erano state acquisite sia software che hardware sia sistemi operativi che i computer, i personal computer per i dipendenti che dovevano utilizzare questa applicazione e alla fine i conti non tornavano.

Perché?

Perché queste funzionalità non erano quelle richieste, tant'è che se ne usavano la metà.

Un'altra considerazione riguarda alla conclusione dei progetti, cioè la prima torta ci

ha fatto vedere che una volta realizzato il progetto, la gran parte delle funzionalità non venivano utilizzate.

Questa torta invece, ci mostra che sostanzialmente la percentuale di progetti che venivano portati a compimento e che quindi passavano il collaudo finale, fatemi dire è pari a un terzo, il 32%.

Il 44 dei progetti doveva essere pesantemente modificato.

E perché qualcosa non aveva funzionato?

Il 24% veniva cancellati del tutto perché non rispondevano ai requisiti.

Non non non andavano mai in esercizio, questa è una situazione che vedete è stata in qualche modo documentata da questa analisi del 2009, quindi siamo già in epoca di Internet, diciamo, avanzato tra virgolette.

Siamo nel web, 2.0 quindi i numero, la percentuale dei progetti?

Waterford, innovativi in negli ambiti più innovativi, non negli ambiti tradizionali di utilizzo dell'informatica, ma negli ambiti più innovativi.

E del 32% il 32% andavano a buon fine e una percentuale, se ci pensate incredibilmente bassa, vuol dire che i due terzi dei progetti avevano generato dei costi imprevisti o perché andavano pesantemente rifatti.

Rimodificare nel 44% dei casi o perché addirittura venivano cancellati, quindi lavoro, soldi spesi, buttati.

In questo, in questo diagramma, che forse si vede un po piccolino, ma l'importante è seguire un po i colori, vedete tre diagrammi che mostrano sostanzialmente.

E l'andamento teorico di un progetto e il primo lo vedete voi il cursore che si muove, sì.

No, ah perfetto.

 **Ludovico Tortora** 24:20
Sì, sì, sì.

 **Angelica Giroto** 24:21
Sì.

 **Francesco Gentili** 24:22
E.

 **SANTE DOTTO** 24:22

Qui sono sul primo diagramma.

Questo è l'andamento teorico, nasce un progetto, si inizia lo sviluppo e si immagina che questo sviluppo, in un tempo definito, questo è la l'ascissa e l'asse dei tempi venga completato.

Questo è l'andamento teorico, siamo nel mondo del Waterford, siamo nel mondo dello sviluppo lineare.

C'è un punto d'inizio, c'è un punto di termine del progetto?

Bene, cosa succede che in genere cambiavano i requisiti?

Come vi dicevo, quindi, in fase di collaudo a un certo punto viene fuori che c'è da modificare qualcosa.

Vabbè, a parte l'affermazione un po colorita, ma il problema è che di solito quello che era stato sviluppato diciamo che rendeva possibile realizzare un numero di funzionalità minori, quindi sostanzialmente si abbassavano in qualche modo i requisiti utente, si abbassa, si abbassavano i risultati raggiungibili.

Questa operazione però non è che dice, vabbè, ma allora tutto quello che che è fatto funziona, anche se per un numero inferiore di funzioni no?

Perché questo abbassamento dei requisiti comportava una forte rivisitazione di tutta l'applicazione e quindi sostanzialmente ci voleva del tempo e qui siamo sul terzo diagrammi.

Gino qua, per cui rispetto al numero di funzionalità inizialmente richieste, si arrivava a un numero di funzionalità diciamo minore, ma occorreva uno sforzo di modifica di tutto quello che era stato sviluppato, con conseguenti costi e tempi.

Quindi qua vedete il tempo?

Sostanzialmente da questo primo diagramma a questo ultimo è quasi tanto doppiato e le funzionalità sono più che dimezzate.

Questo è un po esasperato, ma vi garantisco non tanto il problema che questi 11 sviluppatori e guru dell'informatica dell'epoca siamo nel 2001.

Volevano affrontare, evitare che succedesse questa cosa e il loro approccio è stato un approccio di questo tipo, è stato un approccio un po alla design thinking, nel senso di dire, ma spezziamo questi obiettivi?

Cerchiamo di sviluppare porzioni inferiori di codice che testiamo man mano che andiamo avanti.

Quindi con una diversa architettura software che consentisse di aggiungere funzionalità in modo incrementale e man mano che ci accorgiamo, casomai che, come succedeva in questo esempio, gli obiettivi vanno rivisti perché le funzionalità

che servono sono anche di meno.

Riaggiusta piano piano il tiro e nella stessa quantità di tempo sviluppiamo le funzionalità realmente tra virgolette necessarie realmente richieste.

In questo modo evitiamo lo spreco di tempo e di soldi che avevamo visto nell'esempio precedente di questo terzo diagramma, quindi la logica di questo modello agile.

E è insita nella parola stessa, agile perché agile, perché non è +1 monolite che viene sviluppato in una modalità lineare senza mai avere un punto di ritorno senza avere la possibilità di riciclare se non alla fine del progetto.

Progetto, ma sono metodi che spezzettano le quantità di codice da sviluppare, diciamo così, in modo da potere ritornare più facilmente a allinearsi con le specifiche utente che eventualmente sono andate man mano modificandosi.

Domande.

Perché dico che è un po' lo stesso approccio insito nel design thinking, perché se vi ricordate uno degli aspetti importanti del design thinking era quello di poter iterare. Avevamo visto a suo tempo un diagramma in cui, ad esempio, tra la fase di defi nella fase di empatia si poteva reiterare ove gli obiettivi definiti nella fase di design non di defin non fossero riconosciuti dagli utenti.

Si ritornava alla fase di empatia e si rimandavano a rivedere i dati e casomai si facevano proposte di obiettivi diversi, così come nelle fasi di sviluppo del del progetto di ideazione del progetto.

Si poteva iterare, magari realizzando dei prototipi verificandoli con gli utenti.

Analizzando i feedback, si tornava magari indietro alla fase di ideazione.

Questo stesso meccanismo in qualche modo viene portato dentro le metodologie di sviluppo dei progetti di sviluppo, in particolare delle soluzioni informatiche.

Proviamo a fare il riassunto New Waterford le attività sono sequenziali, si definiscono le specifiche di dettaglio e su questa base si parte da una stima dei costi e dei tempi di progetto, che diventano impegnativi e che devono essere rispettati a partire da questo contratto in qualche modo stabilito in questa modalità parte il progetto e gli utenti di fatto fino al collaudo non vengono quasi più coinvolti.

Non c'è questo meccanismo di verifica parziale e di analisi dei feedback.

La particolarità?

Se ci pensate irragionevole di questo meccanismo, è che la bontà di quello che è creato, di quello che è sviluppato e misurato sulla rispondenza ai requisiti espressi all'inizio di questo percorso e non è commisurato al valore effettivamente creato.

Le torte ci facevano vedere che del 100 che veniva creato, gli utenti ne riconoscevano utile il 50.

Quindi se noi fossimo andati con una metodologia di sviluppo del progetto, a misurare il valore creato, avremmo dovuto dichiarare un fallimento al 50%, invece si vanno a verificare i requisiti iniziali sui quali non è stata accettata nessuna modifica e si dimostra che tutto quello che era stato chiesto è stato sviluppato.

Peccato che l'utente non l'aveva capito.

Peccato che l'utente, forse in parte non l'aveva neanche chiesto.

Ecco dov'è la scollatura se ci pensate è un paradosso, misurare la bontà del software è sviluppato sulla base dei requisiti espressi mesi fa mesi prima.

Piuttosto che sul valore effettivamente del software creato rispetto alle esigenze degli utenti, che peraltro magari è vero che sono mutate nel tempo, ma questa è una cosa di cui in epoca di trasformazione digitale bisogna tenere conto, perché le stesse piattaforme software e gli stessi software evolvono molto più velocemente, quindi non si può continuare a sviluppare su tecnologie molto innovative, bloccando i requisiti a mesi prima dell'avvio del progetto di realizzazione.

Qual è la conseguenza?

La conseguenza è una serie di sprechi perché si utilizza male il tempo delle risorse umane coinvolte.

E il denaro utilizzato per sviluppare funzionalità inutili e acquisire tecnologie che a quel punto non si sa più manco se servono.

E alla fine tutto viene rilasciato quasi sistematicamente in ritardo rispetto ai tempi inizialmente previsti.

Dunque, quello che in epoca di innovazione spinta deve essere consentito nelle metodologie di progettazione come il Desert thinking del tanto più nelle metodologie di sviluppo come la già il project management.

Rispetto al Watford e la flessibilità nei cambiamenti di contesto, perché i cambiamenti di contesto devono essere gestiti anche durante lo sviluppo del progetto.

La fase di sviluppo non è una fase che si può congelare astraendo dalla realtà che ci cambia sotto sotto i piedi.

Se non si vuole correre il rischio di sviluppare qualcosa che non sempre o non serve completamente, quindi da qui la logica di una metodologia più agile che risponda flessibilmente ai cambiamenti di contesto.

Abbiamo visto l'esempio dello spezzettamento del rilascio nei grafici che vedevamo

prima, adesso entriamo anche sulle logiche, diciamo sulle competenze soft.
Alla base di questo nuovo approccio è qua ritroveremo tanti elementi che abbiamo già visto per quanto riguarda la progettazione, analizzando la metodologia del design thinking, queste queste tre slide che vedremo, di cui analizzeremo un po la prima, poi il resto, se volete c'è la scorriamo, ma insomma è più di dettaglio.
E sul sito del manifesto per lo sviluppo agile del software del 2001, che trovate su Internet esattamente con questo, con questa filigrana di questa immagine che vi sto mostrando.

Che cosa dicono questi signori che sono elencati qua sotto dicono?

Sostanzialmente stiamo scoprendo modi migliori di creare software, sviluppando, sviluppandolo e aiutando gli altri a fare lo stesso.

Grazie a questa attività siamo arrivati a considerazioni importanti.

Allora uno, gli individui e le interazioni ci interessano più che i processi e gli strumenti.

Quindi ascoltiamo le persone.

Il team di sviluppo, gli utenti, i clienti e non blocchiamo voci su piani e processi e strumenti non modificabili, gli individui e le interazioni il software ci interessa che sia funzionante, non ci interessa che ci sia una documentazione perfetta.

Che spiega come gira quel software.

La cosa importante è che funzioni, questo è un punto delicatissimo.

Perché il waterfall aveva un grande vantaggio, perché si prendeva tutto il tempo necessario per era quello di documentare ogni singola riga di codice, quindi un testo in lingua parlata che forse inglese, italiano, la lingua utilizzata che descriveva esattamente i vari moduli software come interloquiva tra di loro e quali funzionalità erano sviluppate da ciascun modulo software.

Per cui quando si doveva andare a fare manutenzione del codice non si perdeva tempo a rileggersi il linguaggio con cui era stata scritta quella applicazione, ma si prendevano i manuali della documentazione e nella lingua che si parlava si capiva come girava quel codice.

Qual era il modulo e la riga di codice da andare a toccare?

È giusto avere maggiore attenzione al software funzionante, è un po quella logica del beta test che dicevamo ti rilascio il software e tu utente mi dici se funziona o come modificarlo.

Però è anche vero che uno dei rischi nello sviluppo ha Gil del software e che si trascuri troppo la documentazione e quando poi ci si devono rimettere le mani è un

po un problema.

Quindi attenzione perché diciamo, queste affermazioni non vanno prese come o tutto bianco o tutto nero e un'attenzione maggiore.

Consideriamo importanti più che il software funzioni piuttosto che ci sia tutta la documentazione esaustiva sviluppata e il rilascio della perché questa sottolineatura che sembra quasi un po' contraddittoria, un po' sbagliata perché spesso non si rilasciavano le applicazioni.

Perché erano state scritte le le perché la documentazione non era completa, non era stata iscritti manuali, per esempio le grandi, le grandi Eh.

In società di sviluppo del software come IBS, ad esempio in quegli anni non rilasciava i la versione dei prodotti lingua italiana fintanto che non era stata sviluppata tutta la traduzione in lingua italiana.

Quindi diciamo tempi molto lunghi di rilascio delle applicazioni.

Ora dov'è il problema?

Il problema è che se si corre troppo si trascura la documentazione, poi il dazio si paga nel momento in cui bisogna metterci le mani, però loro non stanno dicendo non documentate i programmi, loro qui dicevano, bisogna considerare importanti il software funzionante più che la documentazione esaustiva, quindi è un accento che viene posto su un qualcosa che invece, sbagliando, veniva trascurato.

Il rilascio di un software funzionante.

Un'altro aspetto da considerare importante la collaborazione col cliente, più che la negoziazione dei contratti, la negoziazione dei contratti vuol dire quando parte il progetto di realizzazione definiamo le specifiche utente, quello è un documento che viene sottoscritto dagli utenti e dagli sviluppatori ed è un contratto tra gli utenti e gli sviluppatori, per cui è come se gli utenti si impegnassero ad accettare quello che gli sviluppatori svilupperanno a fronte di quel contratto senza nessuna modifica possibile, quindi capite bene che la collaborazione col cliente e l'accetta l'accento sulla collaborazione del cliente vuol dire anche nel momento in cui cambiano.

Cambiano i requisiti del cliente, quindi quel contratto negoziato una volta sulla base dei requisiti utenti sottoscritti all'inizio non è un contratto vincolante che non può essere più modificato.

Non siamo nel mondo del water, non vogliamo più essere nel mondo del Waterford. Vogliamo porre l'attenzione a rispondere al cambiamento piuttosto che rispettare un piano che magari rilascia qualcosa.

Passati mesi, forse anni, nei progetti più importanti che è già superato

dall'evoluzione delle tecnologie digitali, che nel frattempo sono modificate, questi sono gli accenti e questa sottolineatura che secondo me doveva essere scritta in grande.

Queste due righe prima dell'elenco dei nomi dice, ovvero, fermo restando il valore delle voci a destra, quindi esattamente quello che vi dicevo prima, consideriamo più importanti le voci a sinistra.

Come dire, non potremo mai sacrificare la funzionalità del software alla esaustività della documentazione, non potremmo mai sacrificare la risposta al cambiamento, al congelamento e al rispetto di un piano che nel frattempo è diventato obsoleto perché son cambiate le condizioni di contesto.

Questo è il manifesto.

Ora piccola parentesi, non vi chiederò mai se qualcuno siete stesse preoccupando in questo momento.

Io lo dico perché?

Perché la logica di questo e di questo corso EE magari è diversa da tanti altri e quindi ve lo ripeto fino alla nausea, scusatemi, ma voglio essere chiaro, qui non ci dobbiamo.

Non non dovrete imparare a memoria ste cose, non le chiederò mai, non ve le chiederemo mai all'esame.

Per capirci, è importante che voi però ricordate le cose più importanti, ricordate la logica che c'è dietro, quindi io vi invito queste queste slide a leggerle e rileggere un paio di volte.

Perché voi possiate un po metabolizzare le cose che stiamo raccontando, perché è una prima lettura, sembrano tutte scontate, ma se ci riflettete un attimo.

Noi, per esempio stasera stiamo discutendo di un cambiamento che è stato negli anni dal 2000 a oggi epocale per il mondo delle tecnologie digitali.

Perché cambiare queste modalità di sviluppo dei progetti?

Ancora oggi è un problema perché magari si va a utilizzare la modalità agile per sviluppare progetti che non hanno bisogno della della modalità.

Già il e si fa casino oppure si usa la modalità gilè e non fermo restando il valore delle voci a destra, ma dimenticandosi completamente delle voci a destra e alla fine si fa un gran casino uguale.

Quindi siamo ancora, secondo me, nel pieno di questa trasformazione.

Con un fenomeno anche molto complesso che deriva dalla evoluzione velocissima delle tecnologie digitali.

Non si fa tempo a iniziare un progetto che già le tecnologie sottostanti sono cambiate.

Per certi versi, quindi noi stiamo vivendo un'accelerazione.

Di tutte queste logiche.

E di fatto gli sviluppatori inseguono una realtà che muta a una velocità apparentemente e talvolta veramente insostenibile.

Vabbè, i principi sottostanti per il risparmio perché sono soltanto altre frasi che dettagliano meglio quei quei punti del del del diciamo di obiettivi, insomma.

Qui entrano un po più nello specifico, se avete voglia vi leggete, ma insomma, se volete poi ci torniamo.

Intanto volevo chiudere un attimo il ragionamento, allora.

Da questi principi sono venuta fuori una metodologia di sviluppo del software, ma. Astruendo dal tema dello sviluppo software si è anche definita una modalità di sviluppo dei progetti, indipendentemente dal fatto che fossero progetti informatici e una modalità applicabile in qualunque contesto, cioè il mindset che è sotto queste fasi.

Definite anche qui.

Vedremo che sulla base di questi principi sono state definite varie metodologie, un po come il design thinking.

No, ognuno poi ogni società di consulenza, ogni consulente, quasi ogni scuola di pensiero, si è costruito un po il suo framework.

La stessa cosa e successo per lo sviluppo agile, ci sono diversi framework, in particolare vi ho preparato un documento di qualche slide che riguarda il framework più utilizzato nella trasformazione digitale delle organizzazioni che il framework.

Scrum, Scrum e la mischia del rugby.

E da questo questa immagine della mischia del rugby che funziona in un certo modo, nasce una schematizzazione, un framework che identifica una serie di fasi.

È una modalità per realizzare queste fasi di un processo di sviluppo agile, ma i punti chiave di qualunque framework sono questi.

Adesso te li vediamo insieme, in parte li abbiamo già toccati.

Primo punto, l'elemento organizzativo, profondo, mentale, primario e il team, il team deve essere il più snello possibile, ma deve avere dentro tutte le competenze necessarie.

Quindi all'interno di un team, magari ci devono essere persone necessarie, che ci siano persone che hanno competenze specifiche, alcune competenze sui linguaggi,

alcune competenze sulle architetture, alcune competenze sulla materia trattata. Quindi i team devono essere snelli ma focalizzati, cioè con competenze diverse e focalizzati all'obiettivo.

Tutte le competenze necessarie per realizzare l'obiettivo, minimizzando le interdipendenze esterne.

Quindi un team quando ha capito che cosa va fatto insieme agli utenti insieme agli stakeholder, insieme ai committenti, si chiude sviluppa quella componente, quel pezzetto.

Attenzione, quel pezzetto la modalità di lavoro di questi team è una modalità di auto organizzazione, quindi non vale la gerarchia organizzativa nelle elezioni che faremo col professor D'ORSI a partire da da dalla prossima, dalla prossima settimana vedremo proprio l'evoluzione organizzativa delle imprese.

E questa logica della consistenza dal punto di vista delle competenze e della capacità organizzativa autonoma dei team risponde a una logica organizzativa che vedremo. Viene chiamata letteratura di tipo olistico, cioè un'organizzazione che si disegna su entità che costituiscono organismi in grado di funzionare al loro interno e di buttare fuori e unità di lavoro consistenti e utilizzabili dal resto dell'azienda.

Questa è una cosa importantissima, siamo fuori dal discorso gerarchico.

All'interno del team le persone sono.

C'è un primus Inter pares che project manager che fa una serie di cose ma che fa una serie di cose, è come dire utili, necessarie nel rispetto di tutti i componenti del team di progetto.

Il lavoro è suddiviso in interazioni in blocchi, piccoli blocchi che si chiamano sprint, nel per esempio nella metodologia SCRUM, nel framework SCRUM.

Perché abbiamo detto, non si sviluppa tutta l'applicazione, si sviluppano pezzetti consistenti dell'applicazione quindi e componenti e che consentono di sviluppare iterando parti del progetto da realizzare.

Al termine di ogni sviluppo di ogni sprint, si possono riverificare le funzionalità ed eventualmente ridefinire le priorità.

Quindi è un po come il motto del del del Design thinking e vi ricordate?

Sbagliare presto e spesso, cioè non frenarsi nella progettazione.

Lì si diceva bloccando delle idee, ma essere pronti a cambiarle il più presto possibile se si dimostrano errate, riciclando, iterando e quanto lo stesso la stessa questione.

Per realizzare software di valore bisogna possibilmente verificarlo in corso d'opera, attraverso prototipi che siano rapidi da sviluppare.

EA basso costo e il feedback dato dagli utenti l'approccio delle piattaforme che dicevo prima vi rilascio delle beta test e la raccolta dei feedback dati dal mercato o dai diversi stakeholder che ruotano attorno all'organizzazione, in cui si stanno sviluppando questi progetti è il motore fondamentale dell'apprendimento e permette al team di trovare la strada migliore per raggiungere il suo obiettivo. Quindi ogni sprint, ogni pezzo viene collaudato, viene in Laboratorio, viene rilasciato, viene come se fosse 1 1 pezzo, un prototipo esso stesso e si analizzano i feedback e si decide come cambiare, se cambiare, come andare avanti all'interno del team, le comunicazioni devono essere, devono viaggiare alla velocità della luce, frequenti e simmetriche.

C'è tutta una.

I malevoli la chiamano liturgia di cecca, addirittura quotidiani.

Sull'avanzamento del progetto che gli sviluppatori fanno in tempo reale online, anche considerate che sono time di sviluppo di gente che possono stare in parti diverse del mondo.

Mondo Eh, perché ormai si sviluppa sostanzialmente in rete i team.

Ecco un'altra questione molto importante, sono le riunioni di e di verifica di quello che è successo, di analisi retrospettiva, di quello che è successo nello sviluppo di quella componente, in modo che ci si interroghi anche su come più e in modo più efficiente, sviluppare la successiva.

Quindi queste sessioni di retrospettiva sono fondamentali, perché?

Consentono un po, nella modalità che vi dicevo prima della qualità totale della Toyota, di trovare il tempo per capire come lavorare meglio.

Fermarsi un attimo, rilasciate uno sprint, capire cosa è successo è cercare di evidenziare quello che si può migliorare in modo da non fare gli stessi errori andando avanti con lo sviluppo.

Quindi questa ricerca del miglioramento continuo.

Ecco, su questi principi voi secondo me dovete rileggervi, non tanto per ricordarvi a memoria, anche qui nessuno vi chiederà mai l'elenco, ma per entrare un po dentro questo spirito, perché poi sono punti che come vedete, sono molto collegati in uno all'altro.

Perché l'ultimo punto si può fare se tutti gli altri in qualche modo precedenti, sono stati indirizzati.

E quindi alla fine viene fuori un'attenzione quasi automatica, diciamo così, alla qualità del lavoro svolto e alla sua rispondenza alle ai requisiti dell'utente e quindi al valore

che viene portato agli utenti, non tanto al rispetto banale tra virgolette, perché poi non si riusciva neanche a fare, tra l'altro di un piano di lavoro.

Se ci sono domande fermatemi Eh.

Se ci sono punti.

Quindi, anche in questo caso ci troviamo di fronte, come esattamente abbiamo detto, per la progettazione, a un cambiamento di mentalità, a un nuovo manager, a un nuovo, un nuovo mindset, a nuovi stili di management, a nuovi stili di leadership all'interno di un gruppo di lavoro, quindi da una leadership basata sul Command control, a una leadership basata a sul servizio che ciascun componente del team offre agli altri componenti del team.

Per questi motivi la Gil si è fatto strada, come è scritto qui nell'ultima frase, nei contesti di reale trasformazione digitale delle imprese, proprio perché più facilmente che le metodologie precedenti hanno consentito di riallineare i tempi di sviluppo e le funzionalità rilasciate a esigenze degli utenti che potevano modificarsi nel tempo anche durante lo sviluppo del progetto per le mutate situazioni di contesto di mercato, organizzative, tecnologiche.

Vabbè, questo diagramma ci fa soltanto capire che poi dal dal 2010 in poi questi approcci sono stati portati, come dicevamo, sullo sviluppo di iniziative innovative. Non solo dal punto di vista delle tecnologie utilizzati, ma utilizzate, ma anche dal punto di vista organizzativo e di processo.

Quindi è un approccio che oggi è documentato ed è definito attraverso metodologie codificate attraverso framework come si chiamano e in qualche modo rivestono un valore che prescinde, assumerà, diciamo così, i confini dello sviluppo delle applicazioni informatiche, tanto da parlare di processi che indirizzano, che consentono lo sviluppo di in modalità business.

Agility di tutta l'organizzazione.

Vabbè, questo è un po un po un riassunto anche di cose che man mano che raccontavo vi ho in sostanza già detto sull'accelerazione dell'evoluzione digitali, ma anche organizzative sulla crescita della componente di servizio anche nei prodotti più tradizionali che può essere digitalizzata per avere una maggiore efficienza e una maggiore personalizzazione dei servizi offerti.

Offerti che vuol dire vantaggio competitivo?

Lo sviluppo di modalità di interazione con i clienti che avvengono sempre più in rete. Il cambiamento di modalità di lavoro all'interno delle organizzazioni per rendere più efficace il lavoro, quella quello slogan che detto male e lavoriamo meno per ottenere

migliori risultati e cerchiamo di faticare meno per ottenere migliori risultati e il miglioramento dei processi produttivi secondo le logiche delle tecnologie che all'epoca erano emergenti, che adesso sono comuni.

E se to date degli industry 4.0 del mondo, tra virgolette 4.0, questa ve la mostro perché è una cosa carina che se vi rimane in mente la figura vi fa vedere un fenomeno di cui abbiamo parlato tante volte, che è il fenomeno, per esempio, della resistenza che l'essere umano ha al cambiamento.

Allora qui vedete come crescono nel tempo le le.

Le il diciamo le.

Le funzionalità rese disponibili.

Ciao Vittorio.



VITTORIO D'ORSI 55:43

Buonasera.



SANTE DOTTO 55:45

È un piacere vederti.



VITTORIO D'ORSI 55:47

Sono venuto a fare il disturbatore, Buonasera a tutti.



LT Ludovico Tortora 55:51

Buonasera.



SANTE DOTTO 55:51

No, no, se sei venuto in tempo giusto perché da lunedì ti ricordo che se approfitto per ricordarti che sei in Aula.



VITTORIO D'ORSI 55:55

Ehi, me lo ricordo bene, infatti mi sono collegato per vedere dove siamo.
Così miri ambienta.



SANTE DOTTO 56:06

Anche, anche se so che hai già conosciuto praticamente tutti.



VITTORIO D'ORSI 56:12

Absolutamente un bel gruppo, una bellissima classe, sono molto contento.



SANTE DOTTO 56:12

Bene, allora.

Sì, sì, anch'io e no.

Volevo far vedere questo diagramma perché rispetto all'atteggiamento dell'essere umano, al cambiamento, che è un atteggiamento che può essere di di paura, di inizialmente anche di di rifiuto e l'andamento, questa è una legge che voglio dire di queste leggi proprio terra terra che sono fatte non tanto sulla base di modelli matematici ma analizzando la realtà che cambia, come per esempio la legge di Moore.

No, questa è la legge di Moore che dice che il numero di transistor che è possibile stampare su circuito integrato raddoppia ogni 12, 24 mesi.

Questo, tradotto in lingua italiana, vuol dire la potenza elaborativa dei computer.

Nel corso degli anni ha sempre rispettato questa legge, è sostanzialmente raddoppiata ogni 12, 24 mesi e sono diminuiti i costi di produzione.

Per questo oggi possiamo avere computer ad altissime prestazioni a bassissimo costo.

Il nostro, il nostro smartphone non è un esempio lampante, è la la quantità di memoria, la capacità di di connessione nel nostro smartphone non era possibile negli anni 70, ma quand'anche fosse stata possibile avesse, avrebbe richiesto un elaboratore che occupava non so, 5 600 m quadri di un centro di calcolo raffreddato ad acqua, quindi era una roba che nessuno mai poteva immaginare in quegli anni, di avere poi in mano quarant'anni dopo.

Quindi, questa legge empirica, la legge di Moore, descrive come cambia la potenza elaborativa dei dei computer e la legge di Marte, che invece che sta per marketing Technology di questo consulente tecnologo De degli anni 2000, sostanzialmente ci fa vedere cosa avviene nel tempo e nei, confrontando l'evoluzione delle funzionalità messe a disposizione.

Le tecnologie, quindi questa curva esponenziale che sale dà la capacità delle organizzazioni di modificarsi di conseguenza per sfruttarne al meglio le possibilità.

E qui vedete una curva logaritmica.

Quindi, paradossalmente, le organizzazioni hanno una resistenza al cambiamento, un

tema che abbiamo già affrontato, tale per cui al crescere delle funzionalità tecnologiche aumenta il gap che andrebbe colmato per poterne beneficiare appieno. E qui lo vedete perché questa curva sale in modo esponenziale.

Questa tra cresce in modo logaritmico, quindi sostanzialmente più le organizzazioni sono strutturate, più questa fatica aumenta e la storia del passaggio, quando è possibile farlo?

Come dicevamo prima, da una modalità di sviluppo e questo è un tema su cui proprio il professor Torsi tornerà, in particolare nel laboratorio che farete il prossimo anno, giusto?

Vittorio sulla sul project management.



VITTORIO D'ORSI 59:10

Assolutamente si.



SANTE DOTTO 59:12

Ecco questa cosa la vedrete proprio da nel vivo.

Entrerete nel vivo di questo, di questo argomento.

Oggi vi do soltanto questa, diciamo suggestione rispetto agli approcci e che esauriamo in questa singola lezione anche nelle fasi di sviluppo dei progetti informatici, per cui cambiare è maggiormente possibile utilizzando metodologia agile. Perché si può realizzare un cambiamento di un'applicazione con minore fatica, minori costi e in tempi più veloci rispetto ai metodi precedentemente codificati e utilizzati e quindi in qualche modo questo gap tra la capacità di delle organizzazioni di modificarsi per utilizzare al meglio le tecnologie e la velocità di cambiamento delle tecnologie.

Si cerca di tenerlo in in qualche misura sotto controllo.

Va bene, io sulle startup innovative non vi dico granché, perché come potete immaginare tutti questi approcci un po nascono dal mondo delle startup.

Microsoft è stata una startup Google è stata una startup, quindi questo approccio dei dei, dei pionieri del degli sviluppatori di piattaforme in questo mondo che da 2.0 si trasformava in 4.0, è stato poi un approccio non codificato, ma in qualche modo utilizzato sul campo, quindi aziende snelle, focalizzate, native digitali, native digitali, perché sono nate per utilizzare funzionalità che in quel momento erano disponibili in rete?

Eh EE quindi non erano realtà.

Le startup non sono realtà in cui ci sono procedure codificate e burocrazie logiche di potere che in qualche modo si frappongono ai ostacolano.

L'evoluzione chiamiamola organizzativa, che sarebbe necessaria e questa è, diciamo imprinting, questo modo di operare in qualche misura, con questa metodologia gile viene portato su aziende di dimensioni maggiori, sulle pubbliche amministrazioni, sulle grandi imprese, proprio per riuscire a mantenere a realizzare quelle caratteristiche di flessibilità che sono necessarie per indirizzare i cambiamenti di contesto di tutti i tipi e di contesto economico, di contesto organizzativo, di contesto tecnologico che in qualche modo, se affrontate in una maniera coerente, possono dare dei vantaggi competitivi, altrimenti si trasformano in minacce in ostacoli allo sviluppo di quella realtà organizzativa.

Con tutto il tema dei mindset, come dicevamo, e quindi con la possibilità di motivare maggiormente le persone nell'azienda, attrarre talenti, ridurre i costi della burocrazia che ciascuna organizzazione si porta dietro per ovvi motivi, ma tenere sotto controllo questi costi e dei sovraccarichi di lavoro definiti dalle procedure EE dalla?

Definite procedure diciamo stabilite, definite dal management, quindi realizzare un modello più efficace e più flessibile per affrontare la trasformazione digitale.

La quarta rivoluzione industriale, questo è un po il senso di di queste, di queste metodologie sui framework.

Non vi dico granché, perché poi nel caso di studio che che vi lascio magari se volete, lo scorriamo insieme.

Chi ha interesse o perché ci vuole realizzare un approfondimento sopra ai fini del project work?

Perché comunque le magari qualcuno lo sta utilizzando.

Qualcuno di voi che già lavora sta utilizzando tecniche metodologie di questo tipo, se lo può, se lo può riguardare più più diffusamente.

EE invece volevo un po riflettere con voi su questa Carta conclusiva in modo che poi anche nella nella revisione di questa documentazione e del relativo caso di studio teniate presenti questi questi principi.

Allora, sostanzialmente in questo percorso oggi abbiamo visto che, come per la fase di progettazione, quindi il design thinking, per capirci, anche lo sviluppo, quindi la realizzazione di progetti fortemente innovativi.

Basati anche, parliamo di trasformazione digitale sulle tecnologie digitali.

Richiedono sostanzialmente modalità che facilitino la collaborazione tra tutte le parti coinvolte, quindi all'interno del team di progetto del team di progetto all'interno

dell'organizzazione in cui opera e con gli utenti, con i potenziali clienti.

Se siamo un'impresa con i potenziali fruitori, se siamo una pubblica amministrazione, quindi un ragionamento, in sostanza di una migliore collaborazione, di una migliore comunicazione.

La Jail l'abbiamo visto, è nato nel 2001 col manifesto per lo sviluppo agile del software.

È proprio per il fatto di capitalizzare su un approccio su mindset, di maggiore flessibilità, di maggiore apertura dal mondo dello sviluppo del software e si è andato estendendo allo sviluppo di progetti, magari anche non Eh, di sviluppo software.

E in questo contesto della business agility, quindi dell'agilità organizzativa della dell'agilità complessiva e delle organizzazioni che ha forte incidenza sul modo di lavorare delle persone, sugli strumenti di incentivazione, sullo sviluppo delle competenze e su tutta un'altra serie di di questioni che attengono alla allo sviluppo delle soft skill all'interno delle organizzazioni.

Gli aspetti più importanti che abbiamo visto scorrendo i punti salienti del manifesto sono la centralità degli individui e delle interazioni tra loro, il focus sul funzionamento del software e non sulla stesura di una documentazione, diciamo coerente con i requisiti, quindi su andiamo sulla sostanza, non andiamo sulla forma, dove dobbiamo essere molto attenti, ricordare sempre che poi il manifesto adesso dice attenzione perché i punti sulla destra, quindi le parti formali, non è che vanno trascurate perché il software va comunque documentato.

Però l'obiettivo del team di progetto non è documentale.

Un codice l'obiettivo del team di progetto è sviluppare un codice che funzioni e che serva realmente.

Quindi è una questione di focalizzazione, non è una questione di età, non facciamo più alcune cose.

Collaborazione col cliente, capacità di rispondere in modo flessibile ai cambiamenti organizzativi di esigenze di requisiti tecnologici, possibilità di riallineare il codice che si sta sviluppando.

Se stiamo parlando di un'applicazione informatica in modalità il più possibile facile e flessibile per rispondere a questi cambiamenti.

E qui il cambiamento di Mindset Tim snelli che hanno tutte le competenze al loro interno focalizzati sugli obiettivi che si auto organizzano, uno sviluppo software che viene sezionato in componenti più gestibili da un punto di vista, quindi che richiedono minore tempo di sviluppo.

E sui quali si può intervenire più velocemente.

Ancora le.

Nel corso dello sviluppo, se serve, i timbri che si auto organizzano per lavorare in modo iterativo, quindi all'interno del team non funziona la logica gerarchica, ma una logica di competenze che vengono messe in campo nel miglior modo possibile da tutti i partecipanti al team e sviluppando, possibilmente per prototipi, concili di testa e di feedback, cercando di migliorare il processo.

Il processo produttivo alla qualità dei prodotti con queste logiche di retrospettiva, per cui sviluppato uno sprint, cerchiamo di capire com'è andato quello sviluppo. Cerchiamo di fare un po un'analisi di come abbiamo condotto quel lavoro per capire se nello sviluppo successivo nello sprint successivo, magari dobbiamo cercare di modificare qualcosa.

Siamo un po in anticipo, per cui se avete domande eh e riflessioni su quello che abbiamo detto, magari chi di voi avesse mai avuto esperienza di sviluppo e con queste metodologie, se vuole portare la sua impressione, il suo contributo è il momento giusto per farlo.

Anche perché così focalizziamo meglio alcuni aspetti che li ricordiamo meglio, ci riflettiamo sopra, altrimenti questi, questi pochi minuti che mancano.

Vi faccio vedere soltanto che tipo di documentazione vi lascio su Canvas in relazione alla giudice Crumb, framework di cui vi ho brevemente parlato come uno delle degli approcci codificati.

E di utilizzo della della?

Già il project management nell'ambito dello sviluppo di applicazioni.

Se avete domande io, io preferirei che facesse qualche domanda.

Sofia, Dimmi cosa pensi di questa lezione, dai rompi il ghiaccio, se l'unica invidia tu.

Alla fine, arrivati a quest'ora, gli altri zitti zitti si sono tutti scollegati.

Chissà cosa stanno facendo tu sei l'unica seria persona che mi aiuta, che resta in video che mi fa compagnia EE che quindi purtroppo poi alla fine Achille chiedono le cose.

Io lo so che Ludovico è collegato, ma ma non ma non si fa vedere, per esempio, ma anche gli altri, ovviamente Angelica, anche no.



VITTORIO D'ORSI 1:09:14

Stanno cucinando, tante stanno cucinando.

 **Angelica Girotto** 1:09:16
Sì, sì, ci sono.

 **Francesco Gentili** 1:09:16
Ci siamo dovuti, ci siamo.

 **Giovanni Pecoraro** 1:09:18
Andiamo tutti.

 **SANTE DOTTO** 1:09:19
Ecco quindi se avete, se avete anzi guarda Angelica, fa una cosa, fai una cosa, fai una domanda a tutti i colleghi che si stanno collegando in video solo ora.

 **Ludovico Tortora** 1:09:19
Io io mi sto rammentando la coperta mi.

 **SANTE DOTTO** 1:09:29
Così, poi rispondono loro, una cosa che a te non è CHIARA, magari o che vuoi risentirti raccontare così te la raccontano gli altri che hanno ascoltato la lezione. Se proprio siamo in crisi, interveniamo io. Vittorio, sempre che siamo in grado di rispondere alla tua domanda, ovviamente.

 **Angelica Girotto** 1:09:45
Non avrei davvero domande, perché la prima volta che cioè non ho mai affrontato come argomento, non avrei veramente domande da fare, cioè ho sto pensando a qualcosa, ma ma non so effettivamente.

 **SANTE DOTTO** 1:09:59
Ma sai che se c'è qualcosa che ti ha colpito, per esempio, per esempio quando abbiamo un po visto da un punto di vista così teorico ovviamente, perché insomma, ne stiamo discorrendo?
Quindi non.
Ma quei diagrammi, per esempio, in cui si vedevano tutte le problematiche legate allo sviluppo tradizionale del codice rispetto a uno sviluppo a Jail?

AG **Angelica Girotto** 1:10:17

Esatto, sì, sì.

 **SANTE DOTTO** 1:10:18

Tutte, poi se se ti viene in mente qualche qualche qualche domanda su questo o qualche considerazione da fare, oh ragazzi, ma anche voi?

Francesco, Giovanni, Angelica, Ludovico.

Se avete sottolineature, cadere la mia era una battuta.

AG **Angelica Girotto** 1:10:31

Io veramente mi è tutto, tutto chiaro, veramente non avrei nessuna domanda neanche un, neanche un esempio, qualcosa perché effettivamente non avevo mai affrontato prima non lo saprei.

LT **Ludovico Tortora** 1:10:48

E.

Un fare aggressivo.

 **SANTE DOTTO** 1:10:53

È colui che vorrebbe dire qualcosa.

LT **Ludovico Tortora** 1:10:53

Lo.

No, in realtà no, purtroppo io non posso apportare, diciamo cioè non posso portare niente di costruttivo perché noi come ospedale nel senso facciamo poco sviluppo.

Quindi, dal punto di vista del rilascio costante ci sono sicuramente delle cose che, proprio rimanendo squisitamente tecnologici, sono messe prima in test, poi vengono passate in produzione, però non c'è questa cosa di patching continuo.

Alla fine c'è, anzi spesso si comprano dei servizi che vengono aggiornati, anche troppo poco, nel senso che c'è un fornitore che ti dà il macchinario.

Con il programma e alla fine le modifiche.

Anzi, questo è un grosso problema.



SANTE DOTTO 1:11:40

Beh però questa questa cosa adesso non so se se a te succede.



LT Ludovico Tortora 1:11:40

Le modifiche sul sistema operativo le le può fare solo lui.



SANTE DOTTO 1:11:44

Nel caso specifico, però, questa cosa ti potrebbe vedere coinvolto comunque in un in nella situazione di sviluppo di un progetto, magari dal lato del committente dell'utente committente, più che dal lato dello sviluppatore e quindi forse magari comprendi meglio anche vedo che c'è un messaggio in chat, chi è che è intervenuto? Francesco è sparito, il messaggio, Ah Giovanni, e adesso adesso ti lascio la parola.



GP Giovanni Pecoraro 1:12:12

No, io.



SANTE DOTTO 1:12:16

Volevo dire a Ludovico che questi aspetti, comunque che di maggiore coinvolgimento dell'utente del cliente di sviluppo per componenti, magari è un qualcosa che sulla sulle quali tu potrai più facilmente riflettere dal lato del cliente utente che dal lato dello sviluppatore Giovanni, invece tu parlavi di sviluppo.



GP Giovanni Pecoraro 1:12:35

Se io faccio proprio questo di lavoro, praticamente noi praticamente faccio solo un esempio, con un qualsiasi applicativo per la Banca dobbiamo tenerla.

Aggiorno sempre aggiornato perché abbiamo dei delle deadline anche di supporto, abbiamo proprio con tutti i system test la produzione ogni 90 giorni, 80 giorni.

Lasciamo un nuovo, per esempio, base di teams.

Questo perché per tenere sempre aggiornato il parco macchine, noi facciamo dei controlli di scripting appositi per filtrare, reinstallare con la versione più recente coloro che hanno proprio aggiornamento.

Chi non attiva il PC spento, chi tipo e sospende sempre gli aggiornamenti oppure per i nuovi PC che quando vengono installati esatta, magari una base su sto cane, scusate?



SANTE DOTTO 1:13:17

Però ci si sente, si sente.



Giovanni Pecoraro 1:13:19

No, stavo dicendo, e quando?



SANTE DOTTO 1:13:22

A meno che non ti stesse mordendo il cane, allora, in quel caso, per carità, Interrompi
Eh.



Giovanni Pecoraro 1:13:27

No, no, non volevo disturbarti comunque.



SANTE DOTTO 1:13:30

Però, da quello che capisco, Giovanni, tu se ho capito, Eh aiutami a comprendere tu
lavori in un team che si occupa di gestione e manutenzione.



Giovanni Pecoraro 1:13:39

Gestione, manutenzione e sviluppo, perché ci sono anche i pacchetti di mercato?



SANTE DOTTO 1:13:41

Ha anche sviluppo.



Giovanni Pecoraro 1:13:43

Abbiamo, ma abbiamo anche i pacchetti Custom che proprio creiamo noi da zero,
come la VPN, cioè della VPN.



SANTE DOTTO 1:13:47

Ah.

Perfetto, perfetto, ho capito, ho capito.



Giovanni Pecoraro 1:13:51

Quelli, quelli mercati sono Di inps.

Di break si scoprono simili questi pacchetti, che comunque bisogna manutenzione

soprattutto per via delle nuove macchine, nuove macchine.

L'intera San Paolo, hanno un kit di installazione che viene aggiornato ogni sei mesi.

Però noi quando passiamo alla distribuzione produzione dobbiamo far sì che filtriamo questi PC perché sennò hanno una base vecchia che non gli funziona.

Quindi l'utente che si collega per la prima volta questo PC avrebbe un team che non funziona, può comunicare e con lo smart working.

Adesso la settimana corta, che intesa?

Diciamo che serve molto.



SANTE DOTTO 1:14:23

Eh certo, certo sì, sì, quindi insomma, in queste logiche mi sembra di capire che ti ci sei coinvolto appieno proprio.



Giovanni Pecoraro 1:14:30

Eh?

Il 90% del mio lavoro è proprio questo.



SANTE DOTTO 1:14:34

Quindi una vittima più che altro di tutto quello che ho raccontato.

Insomma, esatto, Angelica, volevi aggiungere qualcosa?



Giovanni Pecoraro 1:14:38

Esatto.



Angelica Giroto 1:14:45

No, no, tutto sì sì.



SANTE DOTTO 1:14:46

Tutto a posto, va bene.



VITTORIO D'ORSI 1:14:49

No, considerate se posso.

Sante il fatto che la diciamo la la la differenza fondamentale è proprio sempre il il requisito utente, cioè nel momento in cui noi abbiamo un committente che ha molto ben chiaro dove vuole andare, ci possiamo permettere un tipo di sviluppo a cascata?



SANTE DOTTO 1:14:52

Come no?



VITTORIO D'ORSI 1:15:17

Perché cosa succede in quel caso l'utente ha una visione chiara delle proprie regole e quindi ehm, anche di come in qualche modo le vorrebbe implementare, quand'è che ci troviamo in situazioni del genere?

In realtà molto molto strutturate, per esempio in algoritmi bancari piuttosto che di polizze assicurative.

Non sto parlando dell'applicazione rilasciata per l'utente Eh sapete la la banca tutta intorno a te sto parlando di spesso e volentieri software di funzionamento di back Office, per esempio le provvigioni degli assicuratori.

Il calcolo è noto, cioè l'assicurazione sa che quegli assicuratori che fanno ancora le polizze stampate, eh, hanno delle regole molto chiare e molto prestabilite e quindi il nostro committente ha una serie di regole note, nitide, precise.

EA quel punto faremo uno sviluppo a cascata molto diverso, sempre nell'ambito assicurativo, quindi pensate la differenza di focus, no a seconda di chi è l'utente mi si sposta.

L'approccio anche progettuale e invece rilasciare servizi innovativi ad un utente finale, per esempio il fatto di poter.

Sospendere la polizza assicurativa online direttamente dal telefonino piuttosto, e farlo nel modo più semplice possibile, non necessariamente compilando Carta.

E allora lì mi si sposta completamente e mi serve un approccio agile fatto con wave, dove alla prima ondata faccio un rilascio semplice della APP, alla seconda, alla terza ondata vado a fare il rilasci per affinamenti successivi in modo tale che riesco a sondare anche col mio utente finale come sta andando questa applicazione?

E vado a fare delle tarature?

No i.

Il fine tuning delle tarature successive e cercando di far sì che questa applicazione, questa APP, che magari è sulla sul telefonino, sia molto più vicina all'esigenza all'utilizzo dell'utente.

Quindi immaginate, stesso committente?

Grande gruppo assicurativo, da un lato fa un software per calcolare le provvigioni dei propri venditori e quello è molto stabile.

Le regole sono fisse e valgono così, magari da anni, dall'altro mi fa un software molto molto veloce, dove cerco di fare passarmi il termine l'acchiappo utente.

No a cui serve veramente una polizza assicurativa veloce, anche quella del tipo sospendere riattivo in in giornata perché non uso la moto?

Tutti i giorni e scendo in garage e mi riattivo l'assicurazione.



SANTE DOTTO 1:18:40

Ma anche perché il Vittorio, una volta che l'utilizzo di quest'APP si è diffuso, è che si parla di settimane, non di anni.



VITTORIO D'ORSI 1:18:40

Molto.



SANTE DOTTO 1:18:49

Magari ti ci metto, butto De ti ci butto dentro perché nel frattempo ho visto che tu hai una certa età, probabilmente hai una casa da gestire, eccetera.

Anche l'offerta per l'assicurazione casa, quindi diventa tutto un meccanismo di implementazione graduale per blocchi successivi, ma continua nel tempo di servizi proprio che indirizzano quello che diceva Vittorio e anche esigenze non espresse da parte degli utenti, se ci pensate.



VITTORIO D'ORSI 1:19:18

Esatto.



SANTE DOTTO 1:19:18

Che magari hanno successo, perché poi magari le polizze casa la sottoscrizione di polizze casa aumenta anche in virtù di questa possibilità di fare.

Ma marketing in rete un certo senso.



VITTORIO D'ORSI 1:19:31

Guardate una volta sono stato o ero a Milano, Eh, per un master e ed è venuto il team di pronto pro pronto però è anche questa.

Poi è un'APP, ma in realtà era un sistema dove si censiscono professionisti di tutte le categorie per le esigenze più disparate.

A Roma va un po di meno, a Milano funziona molto di più, mi serve l'idraulico, vado

sul pronto pro, c'è la APP.

Chiedo il preventivo, mi arriva entro un'ora e in due ore ho l'idraulico in casa, OK il.

Il team di prontopro lavora sostanzialmente sempre, perché ogni volta con una wave portano dentro magari un'altro tipo di gruppo di professionisti.

Professionisti sono partiti da quelli che facevano i lavori internamente a alla casa, sostanzialmente, e poi piano piano hanno iniziato a portare dentro coach psicologi e che non erano categorie che loro inizialmente avevano contemplato.

Pronto, bro, nasceva esclusivamente per i lavori espressi veloci da da fare in casa, quindi immaginate anche loro quanto hanno dovuto virare creando sviluppi completamente nuovi e l'hanno fatto ad ondate successive.

Che si sono resi conto che tantissimi professionisti si sono censiti sulla loro piattaforma e gli utenti allo stesso modo hanno fatto grande richiesta e grande utilizzo.



SANTE DOTTO 1:21:17

E tra l'altro, Vittorio, l'esempio che tu fai è importante anche dal punto di vista della gestione dei feedback.

È una roba che non c'entra niente con applicazioni informatiche.

Vittorio ha fatto un esempio di un processo informatizzato gestito in rete, ma la gestione dei feedback è fondamentale, perché se io mando in giro gente che invece di riparare il rubinetto, immobilizzano i proprietari e li rapinano e va da sé che c'è qualcosa, la mia piattaforma non verrà mai utilizzata.

Non avrà mai il massimo del punteggio in termini di gradimento, quindi il gestore della piattaforma deve fare anche un lavoro di gestione dei feedback di in qualche modo anche selezione dei professionisti che buttano dentro la rete.

Quindi c'è tutto un processo organizzativo sottostante, anche questo da gestire in modalità veloce e flessibile, è strutturata per arricchire di servizi che portano poi business alla piattaforma.

È un servizio agli utenti che la utilizzano, quindi a me, grazie Vittorio di questo esempio, perché poi parlando di queste metodologie si parte, ovviamente sono nate per la gestione dell'informatica, quelli che si sono inventati l'approccio già è il manifesto.

Erano degli degli informatici, erano dei tecnologi, tu hai fatto giustamente un esempio che prescinde dall'utilizzo, se non per il mezzo utilizzato, se non dal punto

di vista del Canale, diciamo utilizzato, ma non parlavi di sviluppo applicativo?
Parlavi di un servizio offerto AA chiunque.



VITTORIO D'ORSI 1:22:46

Esatto, esatto.

Questo anche per dare l'evidenza di come questo approccio si è veramente evoluto ed espanso, addirittura AA settori molto diversificati anche rispetto a quello strettamente informatico.



SANTE DOTTO 1:23:08

Bene, allora se non avete domande diciamo che puntuali come una cambiale, anche se non si usano più le cambiali, chiudiamo la lezione in orario e vi ricordo che mercoledì faremo una sessione di project work.

Sarà la prima sessione di project work, quindi discussione libera sulle vostre idee progettuali, scelta del del del progetto da realizzare EE ragionamenti anche utili per fare domande su tutta questa parte di corso che abbiamo che abbiamo svolto.

Poi inizierà un'alternanza i lunedì ci saranno le elezioni del professor D'ORSI sulle sul cooperative management, alla gestione delle comunità di progetto.

L'organizzazione che cambia e quant'altro, quindi allargheremo il l'orizzonte della nostra analisi al tema più ampio del del delle organizzazioni in senso lato e vedremo una serie di cose insieme a Vittorio e il mercoledì faremo le sessioni di avanzamento dei project work, quindi discussioni libere, in cui ognuno potrà fare le osservazioni che vuole fare le domande che vuole di scrivere, descrivere magari i temi che sta affrontando, le difficoltà che sta affrontando nell'analisi del project work, in modo che stabiliamo anche una comunicazione tra voi sui lavori che state svolgendo, il. Se vi organizzerete in gruppi in diversi gruppi diversi, si fa per dire, perché siamo pochi.

Insomma, vedo che la frequenza attualmente siamo in 5 o sei, quindi come dicevo l'altra volta possiamo fare al massimo tre gruppi, se se siamo 246, ma se non ne facciamo due, se no ognuno sceglie.

Se chi di voi vuole lavorare per conto suo lo faccia liberamente, non c'è nessun problema, nessun vincolo.

È una riflessione che comunque io vorrei portare avanti anche nelle sessioni in plenaria di questi ultimi mercoledì.

Se ci rimangono quattro lunedì e quattro mercoledì per finire il corso.

Questo è quindi i quattro mercoledì li vorrei anche utilizzare per ritornare un po tutti insieme sulle logiche di quello che abbiamo visto e che arricchirà arricchiremo anche delle cose che nel lunedì precedente magari Vittorio ci ha raccontato, vi ha raccontato e alla fine io mi auguro che che che avrete la necessità, diciamo l'esigenza, magari di rivedere velocemente un po di slide, quelle che vi metteremo già.

Vi ho messo io, vi metterò anche Vittorio su Canvas e confezionare quelle 456 slide del project work.

Venire sicuramente all'esame, perché insomma sulle cose ci abbiamo riflettuto tanto, non è che vi dovete stare a imparare a memoria.

Molto.

La cosa che vi invito a fare è riscorrere queste slide che sono molto ricche ma io esempi di quasi di tutto quello che volete ma la cosa importante fissate la vostra attenzione e riflettete sulle questioni più rilevanti dal punto di vista dell'approccio, dal punto di vista del mindset.

Non tanto sì, vabbè.

Le singole fasi deliverable va bene, magari è importante, ma la cosa più importante è ragionare su sulla modalità con cui si deve interpretare questo processo di trasformazione digitale delle organizzazioni.

E ci siamo tornati a cerchi concentrici e faremo anche un'altro giro con con le elezioni di Vittorio sull'organizzazione e quindi abbiamo cercato di analizzare questo fatto da diversi punti di vista.

Eh?

Ed è importante, penso da da parte vostra, in qualche modo concentrarsi un po sulla sul senso che hanno queste modalità EE su perché sono necessarie e su perché sarà utile, dal vostro punto di vista.

Se come mi auguro, andrete a lavorare su progetti importanti di trasformazione digitale delle organizzazione per queste competenze soft questi approcci, metterli realmente in campo e vi garantisco che non sarà facile perché sembra facile dire facciamo squadra, ma poi le dinamiche in una squadra sono impegnative EE veramente bisogna partire carrozzati perché ci abbiamo riflettuto bene perché altrimenti la prima cosa che viene da fare dice vabbè, ma sta cosa è troppo faticosa? Sì, capacità di ascolto, ma io non c'ho tempo, non c'ho pazienza, dicessero un po quello che gli pare e non funziona.

E in questo modo non funziona, per cui la riflessione, secondo me su questi, su

queste logiche, su questi approcci, è la cosa più importante che noi con Vittorio l'abbiamo progettato così questo corso perché vi rimanga qualcosa di utile per il futuro.

Che poi noi l'abbiamo vissuto sulla nostra pelle prima di voi, quindi non è che vi stiamo raccontando cose lette sui libri e basta.

L'abbiamo anche lette sui libri, qualcuno Vittorio l'ha pure scritto, ma è questione di vita vissuta.

E siccome speriamo, pensiamo che voi siete la generazione rappresentata dalla generazione che deve al meglio gestire questa transizione digitale, perché siete quelli ormai che state entrando e siete già entrati in parte nel mondo del lavoro è questa questa cosa?

Siamo convinti che se vi attrezzate bene dal punto di vista delle competenze software, della capacità di riflettere criticamente su quello che ci gira intorno, vi aiuta e ci aiuta in generale ad avere una trasformazione che sia coerente e che tenga veramente al centro le persone EE che colga tutte le opportunità evitando i rischi connessi.

Perché poi è questo il punto e per cui ecco, speriamo di avervi UER così, dato un po' un'idea di quali potrebbero essere i sentieri da seguire.

Ecco, mettiamola così.

Detto ciò ho finisco sennò finiamo tardi.

Vi ringrazio per l'attenzione, grazie Vittorio per avere partecipato, noi ci vediamo mercoledì, dopodomani come.



VITTORIO D'ORSI 1:29:01

Grazie per.

Grazie a tutti, grazie a tutti per.



SANTE DOTTO 1:29:08

Grazie a te e ci vediamo mercoledì stessa.

Ora stesso canale.



SOFIA SECCHI 1:29:14

Arrivederci.



SANTE DOTTO 1:29:15

Ciao a tutti, buona serata grazie a voi.



VITTORIO D'ORSI 1:29:15

Buonasera.



LT Ludovico Tortora 1:29:15

Grazie 1000, arrivederci, buona serata, Ciao a tutti.



FG Francesco Gentili 1:29:16

Buona serata grazie 1000 buona serata.



SS SOFIA SECCHI 1:29:17

Grazie.



SANTE DOTTO 1:29:23

Voi Vittorio, eccoci una bella classe come.



VITTORIO D'ORSI 1:29:24

Ciao santi, interrompi la registrazione un secondo.

Interrompi la registrazione un secondo.



SANTE DOTTO 1:29:31

Ah si, certo brava, dicevo veramente una bella classe.

● **SANTE DOTTO** trascrizione arrestata