

TMMi nel mondo Agile

Versione 1.4

Prodotto dalla TMMi Foundation

Autore: Erik van Veenendaal

Avviso di copyright
Distribuzione illimitata soggetta a Copyright
Copyright © TMMi Foundation, Irlanda.

Il materiale della TMMi Foundation è fornito "così com'è".

La TMMi Foundation non fornisce alcun tipo di garanzia, espressa o implicita, in merito a qualsiasi questione, comprese, ma non solo, le garanzie di idoneità allo scopo o di commerciabilità, di esclusività o di risultati ottenuti dall'uso del materiale. La TMMi Foundation non fornisce alcun tipo di garanzia in merito all'assenza di violazioni di brevetti, marchi o copyright.

L'uso di qualsiasi marchio commerciale in questo documento non intende in alcun modo violare i diritti del titolare del marchio.

Si autorizza la riproduzione di questo documento e la preparazione di opere derivate per uso interno, a condizione che le dichiarazioni di copyright e di "assenza di garanzia" siano incluse in tutte le riproduzioni e opere derivate.

Le richieste di autorizzazione a riprodurre questo documento o a prepararne opere derivate per uso esterno e commerciale devono essere indirizzate alla TMMi Foundation.

TMMi[®] è un marchio registrato della TMMi Foundation.

Hanno contribuito

Asim Ali	(Emirati Arabi Uniti)
Katalin Balla	(Ungheria)
Clive Bates	(REGNO UNITO)
Jan Jaap Cannegieter	(Paesi Bassi)
Vahid Garousi	(Paesi Bassi)
Alon Linetzki	(Israele)
Fran O'Hara	(Irlanda)
Jurian van de Laar	(Paesi Bassi)
Leanne Howard	(Australia)
Poonam Jain	(India)
Tim Moore	(REGNO UNITO)
Alfonsina Morgavi	(Argentina)
Meile Posthuma	(Paesi Bassi)
Matthias Rasking	(Germania)
Chaobo Shang	(Cina)
Erik van Veenendaal	(Bonaire - Caraibi Paesi Bassi)
Blaine Webb	(REGNO UNITO)
Karolina Zmitrowicz	(Polonia)

Traduzione in Italiano

Salvatore Reale	(Italia)
Giorgio Pisani	(Italia)

Storia della versione

Questa sezione è fornita solo a scopo informativo.

Versione	Data	Commento
1.0	30/06/2017	Capitolo introduttivo e tutte le aree di processo TMMi Livello 2 affrontate
1.1	16/05/2018	L'interpretazione in un contesto Agile di tutte le aree di processo TMMi di livello 3 aggiunte.
1.2	10/12/2018	L'interpretazione in un contesto Agile di tutte le aree di processo TMMi di livello 4 aggiunte.
1.3	03/07/2019	L'interpretazione in un contesto Agile di tutte le aree di processo TMMi di livello 5 aggiunte.
1.4	24/12/2019	Aggiunte le sezioni 6.4 e 6.5 per indicare esplicitamente che tutti i livelli TMMi 4 e 5 sono applicabili.

Contenuti

1	Introduzione	6
1.1	Obiettivi	6
1.2	TMMi e Agile	6
1.3	Integrazione del Test Maturity Model (TMMi)	7
1.4	Agile	8
1.5	Miglioramento del processo di test in un contesto Agile	9
2	TMMi Livello 2 Gestito	11
2.1	Area di processo 2.1 Politica e strategia di test.....	11
2.2	Area di processo 2.2 Pianificazione dei test.....	13
2.3	Area di processo 2.3 Monitoraggio e controllo dei test	17
2.4	Area di processo 2.4 Progettazione ed esecuzione dei test	19
2.5	Area di processo 2.5 Ambiente di test	24
3	TMMi Livello 3 Definito	26
3.1	Area di processo 3.1 Organizzazione del test	26
3.2	Area di processo 3.2 Programma di formazione per il test	29
3.3	Area di processo 3.3 Ciclo di vita del test e integrazione	30
3.4	Area di processo 3.4 Test non funzionali	32
3.5	Area di processo 3.2 Peer Review	36
4	TMMi Livello 4 Misurato.....	38
4.1	Area di processo 4.1 Misurazione dei test.....	38
4.2	Area di processo 4.2 Valutazione della qualità del prodotto	39
4.3	Area di processo 4.3 Revisioni avanzate	40
5	Ottimizzazione di livello 5 di TMMi	42
5.1	Area di processo 5.1 Prevenzione dei difetti	42
5.2	Area di processo 5.2 Controllo qualità	43
5.3	Area di processo 5.3 Ottimizzazione del processo di test	44
6	Panoramica Applicabilità Obiettivi e pratiche specifiche TMMi	46
6.1	Valutazioni TMMi	46
6.2	TMMi livello 2 Gestito	46
6.3	TMMi livello 3 Definito.....	46
6.4	TMMi livello 4 Misurato	47
6.5	Ottimizzazione TMMi livello 5	47
	Riferimenti	48

1 Introduzione

1.1 Obiettivi

Questo documento spiega come un approccio Agile per lo sviluppo del software combinato con il modello di miglioramento del processo di test TMMi sia un percorso possibile per raggiungere gli obiettivi aziendali. Spiega come TMMi possa essere utilizzato e applicato in modo vantaggioso in un contesto Agile. Vengono fornite alternative comprovate agli approcci di test tradizionali che implementano le pratiche TMMi mantenendo e, forse, anche aumentando l'agilità. Questo documento spiega come TMMi può aiutare le organizzazioni Agile, indipendentemente dal grado di maturità dell'organizzazione, nel loro percorso verso livelli crescenti di agilità, il tutto fornendo un promemoria delle pratiche di testing critiche che spesso perdono visibilità con la crescita delle organizzazioni e l'aumento della pressione sui progetti. Ogni area di processo TMMi e i suoi obiettivi specifici sono discussi uno per uno. Viene illustrato il rapporto con Agile e l'aspetto tipico delle pratiche. Se non ci si aspetta che una pratica venga eseguita in un contesto Agile, poiché non aggiunge alcun valore, questo viene esplicitamente dichiarato.

Molte organizzazioni Agile (piccole) hanno successo e stanno crescendo, ma possono avere pochi processi documentati e un programma di formazione formale per le persone coinvolte. Per mantenere il successo con la crescita dell'organizzazione, è necessaria una maggiore disciplina dei processi. Le organizzazioni hanno quindi il timore di perdere la cultura Agile che ha portato al loro successo. Questa è una delle sfide che si presentano quando si avvia un'iniziativa di miglioramento dei processi di test TMMi in un ambiente Agile.

Questo documento ha molti destinatari; i due gruppi principali sono:

- Organizzazioni **tradizionali mature** che vogliono passare ad Agile mantenendo la maturità dei processi.
- Le **organizzazioni agili** che hanno successo e stanno crescendo. Di conseguenza, hanno bisogno di una sorta di maturità dei processi e allo stesso tempo vogliono mantenere i vantaggi di essere Agile.

Questo documento non è destinato a essere utilizzato indipendentemente dal modello TMMi originale. È inteso a fornire una guida a coloro che eseguono il miglioramento del processo di test in un ambiente Agile sull'interpretazione e il significato dei vari obiettivi e pratiche TMMi. Questo documento integra il modello TMMi originale e deve essere utilizzato come documento complementare.

1.2 TMMi e Agile

La convinzione errata è che gli approcci TMMi e Agile siano in contrasto. Gli approcci Agile e TMMi non solo possono coesistere, ma se integrati con successo porteranno benefici sostanziali. C'è anche la sfida di guardare al testing in modo diverso, essendo pienamente integrato nello sviluppo Agile e cosa significa nel contesto di un programma di miglioramento del "test". Si noti che la "i" di TMMi si riferisce al fatto che il testing dovrebbe essere una parte integrata dello sviluppo del software, e non essere trattato come qualcosa di totalmente separato. La letteratura e le presentazioni sul testing nei progetti Agile tendono a concentrarsi sui test unitari, sull'automazione dei test e sui test esplorativi, ma c'è di più! L'utilizzo del modello TMMi in un contesto Agile fornisce un promemoria delle pratiche di testing critiche che spesso vengono "dimenticate". Questo documento mostrerà con esempi che il TMMi e i metodi Agile possono lavorare efficacemente insieme. La sfida consiste nell'applicare i principi lean per potenziare le pratiche Agile e facilitare le pratiche TMMi.

Quando si implementa il TMMi si deve tenere conto del fatto che l'intento del modello TMMi non è quello di "imporre" un insieme di pratiche a un'organizzazione, né di applicarlo come uno standard a cui si deve "dimostrare la conformità". Utilizzato in modo appropriato, il TMMi può aiutare a individuare le aree di test specifiche in cui il cambiamento può fornire valore in base agli obiettivi aziendali. Questo vale indipendentemente dal modello di ciclo di vita applicato. È importante ricordare sempre che le pratiche TMMi sono una componente attesa, ma possono essere realizzate anche attraverso quelle che vengono definite pratiche "alternative" rispetto a una pratica TMMi definita. Pensate sempre: qual è l'intento della pratica, qual è la logica e come aggiunge valore al business? Spesso in una cultura Agile l'intento è già raggiunto, ma attraverso una pratica alternativa. In genere "qualsiasi" soluzione è conforme, purché sia guidata dalle esigenze di business! Quando si utilizza il TMMi non bisogna essere troppo prescrittivi, perché non è questo l'intento del TMMi. Interpretare sempre gli obiettivi e le pratiche di TMMi in base al

contesto della propria situazione. In generale, stabilendo innanzitutto le esigenze di processo all'interno del contesto aziendale specifico, è possibile prendere decisioni su come concentrare e guidare le priorità di miglioramento dei processi.

La maggior parte dei conflitti che sorgono tra il TMMi e l'Agile si basano su una visione storica del TMMi di come dovrebbe apparire una "buona pratica" quando viene implementata, oppure su un fraintendimento del fondamento logico delle pratiche agili basato sul modo in cui dovrebbero sostenere i valori Agile. Gli esperti TMMi, compresi i valutatori (principali), dovranno riconsiderare e potenzialmente ripensare i messaggi che potrebbero essere inavvertitamente condivisi in merito a come dovrebbe apparire una "buona pratica conforme al TMMi" quando viene implementata. Quando gli approcci Agile vengono implementati in modo appropriato insieme ai processi TMMi, si otterrà un'efficace implementazione delle pratiche di testing, non la loro eliminazione. Si noti che, oltre alle pratiche specifiche (di testing) all'interno del TMMi, esistono anche pratiche generiche. Lo scopo delle pratiche generiche è quello di sostenere l'istituzionalizzazione di un'area di processo, il che significa garantire che l'organizzazione disponga di un'infrastruttura in grado di supportare l'area di processo in caso di ingresso di nuove persone o di altri cambiamenti all'interno dell'organizzazione.

Il passaggio dallo sviluppo software tradizionale ad Agile può anche far emergere l'iniziativa di sfrondare e snellire i processi così come sono definiti oggi. In questo modo le organizzazioni basate su TMMi beneficeranno del modo di pensare Agile. C'è stata la tendenza a leggere nel modello TMMi cose che non ci sono, creando così processi e prodotti di lavoro inutili e senza valore aggiunto. Tornando alle origini e agli obiettivi di miglioramento e utilizzando il modello TMMi nel modo in cui è stato concepito, si favorisce l'allineamento dei processi a reale valore aggiunto con le esigenze e gli obiettivi reali dei processi. La riduzione e lo snellimento dei processi con una mentalità Agile porteranno a processi che riflettono ciò che le persone fanno realmente e garantiranno che vengano raccolti solo i dati che vengono utilizzati. La mentalità Agile porterà anche a concentrarsi sul mantenere le cose il più semplici possibile, cosa che di solito non è facile, ma che porterà un beneficio a coloro che praticano un'implementazione TMMi. I miglioramenti nell'ambito di Agile avvengono tipicamente attraverso piccoli gruppi di lavoro che possono agire rapidamente, e questo è un altro modo in cui TMMi può trarre vantaggio dall'essere Agile. Ricordiamo inoltre che esiste una naturale separazione tra il livello 3 di TMMi e i livelli 4 e 5 di TMMi. Alle organizzazioni particolarmente agili si raccomanda di scegliere e attuare criticamente le pratiche dei livelli 4 e 5 di TMMi che sono importanti e hanno un valore aggiunto. Sebbene il TMMi sia completo, per avere successo le organizzazioni devono identificare le pratiche di testing chiave e i miglioramenti che richiedono attenzione.

1.3 Integrazione del Test Maturity Model (TMMi)

Il framework TMMi è stato sviluppato dalla TMMi Foundation come linea guida e quadro di riferimento per il miglioramento dei processi di test, affrontando le questioni importanti per i test manager, i test engineer, gli sviluppatori e i professionisti della qualità del software. Il test è definito all'interno di TMMi nella sua accezione più ampia, per comprendere tutte le attività legate alla qualità del prodotto software.

TMMi utilizza il concetto di livelli di maturità per la valutazione e il miglioramento dei processi. Inoltre, vengono identificate le aree, gli obiettivi e le pratiche del processo. L'applicazione dei criteri di maturità di TMMi migliorerà il processo di test e ha dimostrato di avere un impatto positivo sulla qualità del prodotto, sulla produttività dell'ingegneria di test e sul tempo di ciclo. Il TMMi è stato sviluppato per supportare le organizzazioni nella valutazione e nel miglioramento dei processi di test.

TMMi ha un'architettura a stadi per il miglioramento dei processi. Contiene fasi o livelli attraverso i quali un'organizzazione passa quando il suo processo di testing si evolve da uno ad hoc e non gestito a uno gestito, definito, misurato e ottimizzato. Il raggiungimento di ogni fase assicura che tutti gli obiettivi di quella fase siano stati raggiunti e che i miglioramenti costituiscano la base per la fase successiva.

La struttura interna di TMMi è ricca di pratiche di test che possono essere apprese e applicate in modo sistematico per sostenere un processo di test di qualità che migliora in modo incrementale. I cinque livelli di TMMi prescrivono la gerarchia di maturità e il percorso evolutivo per il miglioramento del processo di test. Ogni livello ha una serie di aree di processo che un'organizzazione deve implementare per raggiungere la maturità a quel livello. Le aree di processo per ciascun livello di maturità di TMMi sono mostrate nella Figura 1.

Un principio fondamentale del TMMi è che si tratta di un modello generico applicabile a vari modelli e ambienti di ciclo di vita. La maggior parte degli obiettivi e delle pratiche definiti dal TMMi hanno dimostrato di essere applicabili a modelli di ciclo di vita sia sequenziali che iterativi, compreso Agile. Tuttavia, al livello più basso del modello, molte delle sottopratiche e degli esempi forniti sono (molto) diversi a seconda del modello di ciclo di vita applicato. Si noti che nel TMMi solo gli obiettivi sono obbligatori, le pratiche no.

TMMi è disponibile gratuitamente sul sito web della TMMi Foundation. Il modello è stato tradotto in spagnolo, francese e cinese. TMMi è disponibile anche in formato cartaceo (libro).

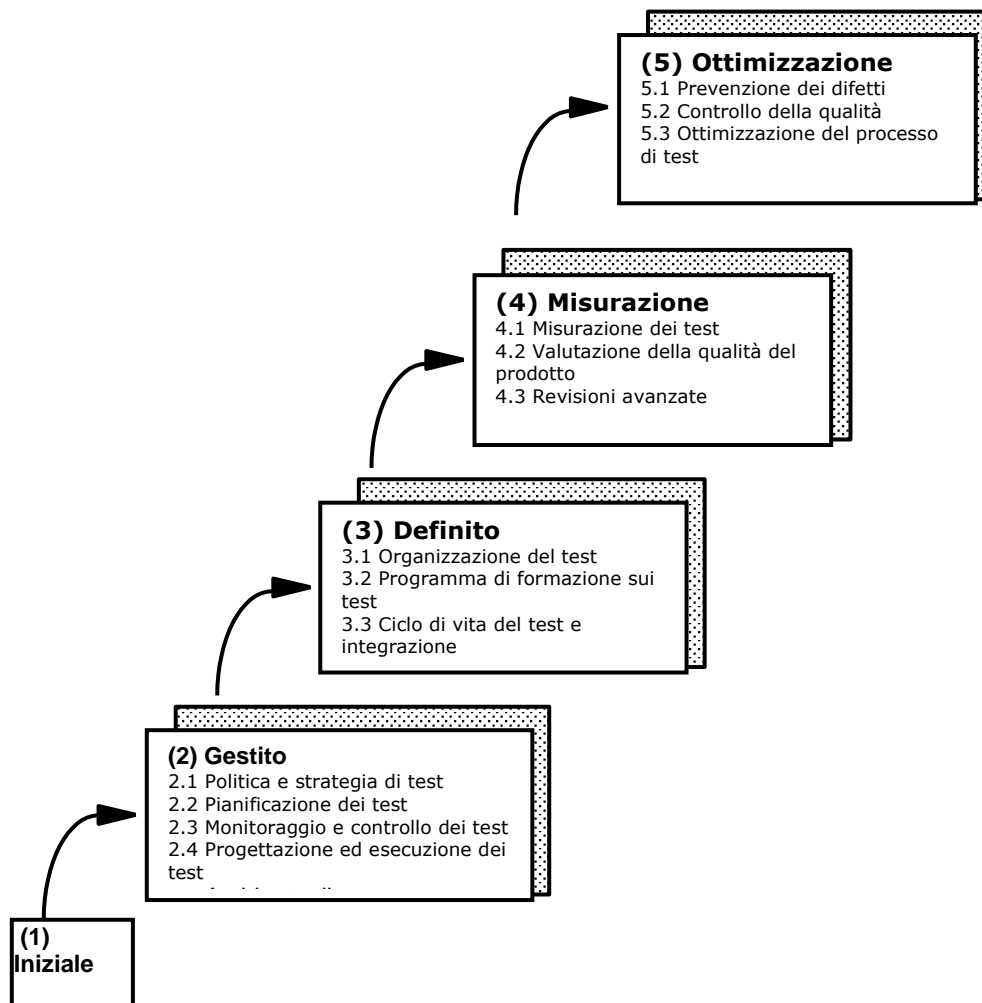


Figura 1: Livelli di maturità TMMi e aree di processo

1.4 Agile

Nel 2001, un gruppo di persone, che rappresentavano le metodologie di sviluppo software leggero più diffuse, ha concordato un insieme comune di valori e principi che è diventato noto come Manifesto per lo sviluppo agile del software o Manifesto Agile. Il Manifesto Agile contiene quattro dichiarazioni di valori:

- Individui e interazioni *più che* processi e strumenti
- Software funzionante e documentazione completa
- Collaborazione con il cliente *rispetto alla* negoziazione del contratto
- Rispondere ai cambiamenti *piuttosto che* seguire un piano

Il Manifesto Agile sostiene che, sebbene i concetti a destra abbiano un valore, quelli a sinistra hanno un valore maggiore. Agile in sé non è una metodologia, ma nel frattempo sono state sviluppate diverse metodologie che

forniscono pratiche per mettere in pratica questi valori. Tre importanti rappresentanti delle metodologie che supportano Agile sono Extreme Programming (XP), Scrum e Kanban.

1.5 Miglioramento del processo di test in un contesto Agile

L'Agile è fortemente incentrato sulla responsabilizzazione dei team, i quali devono eseguire il proprio miglioramento continuo a livello locale, anche attraverso frequenti retrospettive. Alcuni di questi miglioramenti possono essere incentrati sui test. Possono essere indirizzati da un progetto di miglioramento basato su TMMi a un livello superiore, ma possono anche riguardare problemi di processo del team locale relativi al test. È molto importante che l'azione nel contesto del miglioramento del processo di test non venga interpretata e/o utilizzata come una sottrazione di responsabilità locale ai team Agile.

Alcuni dei principali aspetti da considerare, quando si guarda all'influenza di Agile sul contesto di miglioramento, sono i seguenti:

- Frequenza del ciclo di miglioramento
- Aspetti organizzativi
- Portata dei miglioramenti
- Fonte dei miglioramenti
- Livello di documentazione (di test)
- Metodi di miglioramento.

Nei progetti che utilizzano Agile, i miglioramenti avvengono generalmente in frequenti cicli di feedback che consentono di prendere in considerazione frequentemente i miglioramenti del processo di test (ad esempio, alla fine di uno sprint quando si utilizza Scrum). Poiché l'ambito di applicazione è spesso limitato al ciclo o all'iterazione precedente, vengono apportati miglioramenti piccoli ma frequenti che si concentrano principalmente sulla risoluzione di problemi specifici del progetto. Spesso questi miglioramenti non si concentrano sull'apprendimento trasversale e sull'istituzionalizzazione dei miglioramenti.

Guardando a come viene organizzato e gestito il miglioramento del processo di test, è probabile che ci si concentri meno su un gruppo di processo di test a livello organizzativo e si ponga maggiormente l'accento sull'autogestione dei team all'interno del progetto. Questi team hanno generalmente il mandato di modificare il processo di test all'interno del progetto per adattarlo alle proprie esigenze, dando vita a processi altamente personalizzati. Tuttavia, alcune organizzazioni utilizzano anche riunioni settimanali di test "stand-up" per portare le cose a un livello superiore e trasversale al progetto.

Poiché l'attenzione al miglioramento del processo (di test) è più specifica per il progetto, è probabile che venga posta meno enfasi su questioni più ampie che riguardano il testing in tutta l'organizzazione. Ciò potrebbe significare, ad esempio, che i problemi fondamentali del testing potrebbero non essere affrontati appieno perché esulano dal contesto del progetto. Un esempio tipico è l'approccio adottato per testare alcuni attributi di qualità, come le prestazioni e l'affidabilità. Questi problemi possono essere rimandati di iterazione in iterazione perché spesso richiedono competenze e risorse superiori a quelle di cui dispone il team di progetto. In queste aree è difficile fare un passo avanti sostanziale senza grandi investimenti. Risolvere i problemi solo a livello di progetto può anche portare facilmente a una sub-ottimizzazione e a perdere il contatto con il quadro generale.

In un contesto Agile, la gamma e il numero di idee di miglioramento alternative da prendere in considerazione può essere significativamente maggiore rispetto ai modelli di ciclo di vita non-Agile. Poiché tutti i membri eseguono dei test all'interno del progetto, queste idee possono provenire da qualsiasi membro del team. Ciò pone un'enfasi maggiore sulla valutazione e sulla prioritizzazione dei suggerimenti di miglioramento, che può essere più uno sforzo di team che un Task assegnato a un miglioratore del processo di test. Poiché ciò può richiedere la conoscenza specifica del test da parte di un miglioratore del processo di test, questi può anche agire come consulente del team su richiesta.

Nei progetti che utilizzano una metodologia Agile, non aspettatevi di trovare il livello di documentazione di test che vi aspettereste da progetti che utilizzano un ciclo di vita sequenziale. Potrebbe esserci un unico "documento di test" che comprende gli elementi essenziali di una politica di test, una strategia di test e persino un Test Plan di alto livello. Chi migliora il processo di test dovrebbe evitare di proporre "miglioramenti" che richiedono una documentazione di

test più rigorosa e approfondita. Uno dei principi fondamentali di Agile è che la documentazione viene creata solo quando è necessaria in modo chiaro e inequivocabile. Non solo la documentazione di test sarà meno dettagliata, ma lo stesso vale per la documentazione di processo. Spesso negli ambienti Agile si applica con successo una strategia definita "formalizzazione dell'informalità". Se qualcosa funziona bene, non c'è bisogno di cambiarlo per il bene di TMMi. Tuttavia, la documentazione può essere insegnata e condivisa con altri, il che è spesso vantaggioso. Ciò che si intende con la strategia di "formalizzazione dell'informalità" è che se c'è un processo che funziona, ma è informale per certi versi, lo si può insegnare e documentare proprio per come viene eseguito. La leggerezza di un processo implica che non tutti i possibili scenari d'uso sono contemplati nella sua descrizione. Questi processi devono quindi essere supportati da un tutoraggio e da un'assistenza sul posto di lavoro, soprattutto durante il periodo di implementazione iniziale. Di conseguenza, per portare la maturità dei processi di test in un'organizzazione Agile, pur mantenendo la cultura Agile, è necessaria una maggiore - e non minore - formazione. Allo stesso modo, durante l'assessment, l'attenzione per la raccolta delle evidenze si sposterà verso un maggior numero di interviste invece che verso lo studio degli artefatti.

I metodi utilizzati per proporre miglioramenti al processo di test quando si utilizza Agile tenderanno a concentrarsi su metodi analitici per valutare le cause dei problemi, come i diagrammi causa-effetto. Si tratta di metodi particolarmente utili per orientarsi alla risoluzione dei problemi, importante alla fine di un'iterazione.

2 TMMi Livello 2 Gestito

2.1 Area di processo 2.1 Politica e strategia di test

Lo scopo dell'area di processo Politica e strategia di test è quello di sviluppare e stabilire una politica di test e una strategia di test a livello di organizzazione o di programma in cui le attività di test, ad esempio i tipi di test e i quadranti di test, sono definiti in modo univoco. Per misurare le prestazioni dei test, il valore delle attività di test ed evidenziare le aree di miglioramento, vengono introdotti degli indicatori di prestazione dei test.

2.1.1 SG1 Stabilire una politica di test

Qualsiasi organizzazione che intraprenda un progetto di miglioramento del test dovrebbe iniziare con la definizione di una politica di test. La politica di test definisce gli obiettivi generali dell'organizzazione, i traguardi e la visione strategica del test e dei professionisti del test. È importante che la politica di test sia allineata con la politica aziendale complessiva (qualità) dell'organizzazione. I miglioramenti del test devono essere guidati da chiari obiettivi aziendali, che a loro volta devono essere documentati nella politica di test (miglioramento). Una politica di test è necessaria per ottenere una visione comune del test e dei suoi obiettivi tra tutti gli stakeholder di un'organizzazione. Questa visione comune è necessaria per allineare le attività di test (miglioramento dei processi) in tutta l'organizzazione. Si noti che gli obiettivi di test non dovrebbero mai essere un obiettivo a sé stante, ma derivano dall'obiettivo di livello superiore di stabilire la qualità del software e del prodotto funzionante.

Quanto detto è vero anche in un'organizzazione che pratica lo sviluppo software Agile. In effetti, in molte organizzazioni si discute molto del cambiamento del ruolo del testing, dell'indipendenza del testing, dell'automazione dei test e dei tester professionisti nello sviluppo software Agile. Questi e altri argomenti sono tipicamente questioni che dovrebbero essere affrontate in una discussione con la direzione e gli altri stakeholder e documentate in una politica di test. Qualsiasi organizzazione, comprese quelle che praticano Agile, che voglia avviare un progetto di miglioramento dei test deve identificare e definire i driver e le esigenze aziendali per tale iniziativa. Perché altrimenti avviare un progetto di miglioramento? Dedicando del tempo a cogliere le vere esigenze aziendali, si può fornire un contesto per decidere dove concentrare le priorità di miglioramento (del test), ad esempio su quale area di processo. Si noti che una politica di test è tipicamente un documento di una pagina, una pagina web o un diagramma messo sulla parete a livello organizzativo, e non un documento a livello di progetto.

L'obiettivo specifico del TMMi Stabilire una politica di test, comprese le sue pratiche specifiche, è pienamente applicabile alle organizzazioni che applicano lo sviluppo software Agile. Naturalmente gli elementi di una politica di test possono essere incorporati anche in una politica di sviluppo. Non c'è un requisito specifico TMMi che imponga che si tratti di un documento separato. La politica di sviluppo di un'organizzazione che applica lo sviluppo software Agile potrebbe ad esempio identificare Scrum come framework di gestione, XP come metodo principale utilizzato e l'adesione ai valori Agile come principio principale. Un altro principio importante che potrebbe essere citato è che tutti i membri del team sono responsabili di tutto; anche la qualità del prodotto è una responsabilità del team.

Tuttavia per Agile, c'è una grossa attenzione in relazione alla politica di test e in particolare agli obiettivi di test (miglioramento) definiti. Anche se ci possono essere obiettivi generali che riguardano il miglioramento del processo di test nell'organizzazione, questo deve essere bilanciato con i singoli progetti e team Agile che sono responsabili del miglioramento del proprio processo. La sfida del miglioramento dei processi Agile consiste nel guidare e inquadrare il miglioramento a livello organizzativo, senza ridurre il senso di appartenenza del singolo team Agile al proprio processo.

2.1.2 SG2 Stabilire una strategia di test

La strategia di test segue la politica di test e serve come punto di partenza per le attività di test all'interno dei progetti. Una strategia di test è tipicamente definita a livello di organizzazione o di programma. Una tipica strategia di test si basa su una valutazione di alto livello del rischio del prodotto e comprende una descrizione dei tipi di test, dei quadranti di test e dei livelli di test da eseguire, ad esempio: test di unità, di accettazione, di regressione e di

prestazione. Non è sufficiente affermare, ad esempio, che all'interno di un'iterazione verranno eseguiti test unitari e test di accettazione. Occorre definire cosa si intende per test di unità e di accettazione, come si svolgono di solito e quali sono i loro obiettivi principali. L'esperienza dimostra che quando si definisce e si segue una strategia di test, è probabile che si verifichino meno sovrapposizioni tra le varie attività di test, il che porta a un processo di test più efficiente. Inoltre, poiché gli obiettivi e l'approccio dei vari tipi e livelli di test sono allineati, è probabile che rimangano meno buchi, il che porta a un processo di test più efficace e quindi a un livello più elevato di qualità del prodotto. Per stabilire questo allineamento, è altamente consigliabile includere in un'unica strategia di test sia i test Agile che quelli non Agile che si svolgono nello stesso progetto. Tuttavia, se ci sono progetti Agile e non Agile separati, di solito ci sono due strategie di test.

La strategia di test è un documento fondamentale in un ambiente Agile. Definisce, ad alto livello, i test da eseguire nei team Agile (team di iterazione); quali tipi di test, quadranti di test e livelli di test vengono eseguiti e ad alto livello il loro approccio. Il documento descrive l'organizzazione dei test, ad esempio quali test si svolgono all'interno dei team Agile e quali all'esterno. Definisce il rapporto tra i team Agile e i livelli di test che vengono eseguiti al di fuori del loro ambito, ad esempio i test di integrazione hardware/software, i test di integrazione del sistema o i beta test. Una strategia di test assicura che tutti coloro che sono coinvolti nel testing comprendano il quadro generale del testing.

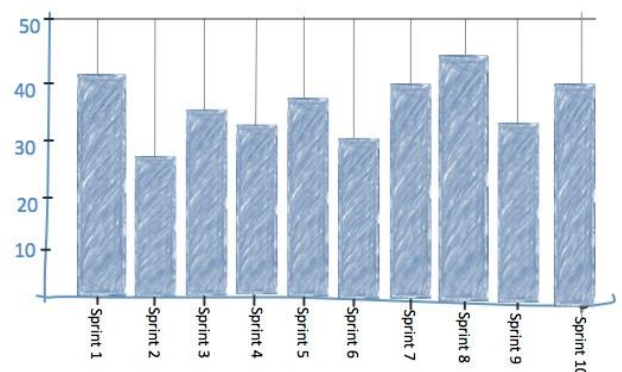
Il documento di strategia di test snella è spesso una buona soluzione per l'organizzazione Agile e una via d'uscita dai Test Plan dettagliati (di progetto). Durante la pianificazione del rilascio, la strategia di test disponibile a livello di organizzazione o di programma viene discussa e confermata, oppure viene creata una strategia di test derivata specificamente per il progetto. La strategia di test confermata o creata fornisce un quadro di test che copre tutte le iterazioni.

L'obiettivo specifico di TMMi Stabilire una strategia di test, comprese le sue pratiche specifiche, è pienamente applicabile alle organizzazioni che applicano lo sviluppo software agile. Si noti che le attività di "distribuzione" attiva, ad esempio per il documento sulla strategia di test, possono essere meno rilevanti in Agile. Chi è parte in causa dovrebbe essere già stato coinvolto nelle discussioni precedenti come parte dell'approccio whole-team. Tuttavia, la strategia di test non è a livello di team, ma a un livello superiore. L'intero team opera a livello di team, quindi un documento organizzativo o di programma deve essere distribuito ai team e ad altri stakeholder, ad esempio a coloro che svolgono attività di test al di fuori del team, se esiste un modello ibrido.

2.1.3 SG3 Stabilire gli indicatori di prestazione dei test

Gli obiettivi aziendali per il miglioramento del test, definiti nella politica di test, devono essere tradotti in una serie di indicatori chiave di performance del test. La politica di test e gli indicatori di performance che la accompagnano forniscono una direzione chiara e un mezzo per comunicare i livelli di performance di test attesi e raggiunti. Gli indicatori di performance devono indicare il valore del test e del miglioramento del processo di test per gli stakeholder. Poiché gli investimenti nel miglioramento dei processi necessitano di un sostegno a lungo termine da parte del management, è fondamentale misurare quantitativamente i benefici di un programma di miglioramento per mantenere la motivazione. Attenzione: questo obiettivo specifico del TMMi riguarda la definizione di un numero limitato (ad esempio, 2 o 3) di indicatori di performance dei test. Non si tratta di impostare e implementare un programma di misurazione completo, ma piuttosto di definire un insieme di indicatori di base che indichino come il valore dei test cambia nel tempo e nei diversi ambienti di fornitura.

Nell'ambito di Agile l'attenzione si concentrerà maggiormente sul team e sul pensiero sistemico. Ciò può comportare un corrispondente allargamento degli indicatori al team e al sistema complessivo, anziché limitarsi alle sole specificità del testing. Anche gli indicatori del livello 2 del TMMi sono principalmente legati ai risultati finali delle iterazioni. Tra gli esempi si possono citare i difetti evasi, la velocità, la valutazione della soddisfazione del cliente, lo sforzo/spreco, la percentuale di automazione dei test, ecc. La sfida consiste nel definire il mix appropriato di indicatori relativi all'approccio



basato sul team e sul pensiero sistemico, fornendo al contempo una buona indicazione dei risultati ottenuti da un programma di miglioramento dei test basato su TMMi.

L'obiettivo specifico di TMMi Stabilire gli indicatori di prestazione dei test, comprese le sue pratiche specifiche, è pienamente applicabile, anche se gli indicatori di prestazione selezionati e applicati possono avere una portata più ampia e più vasta rispetto a quelli relativi ai soli test. Naturalmente quest'ultimo aspetto renderà più impegnativa l'analisi e l'interpretazione degli indicatori di prestazione. Infatti, gli indicatori di performance utilizzati potrebbero non essere chiamati indicatori di performance del test, ma piuttosto indicatori di performance del team o del sistema. Questo va bene nel contesto di TMMi, purché abbia elementi legati al testing e venga utilizzato per valutare i progressi compiuti nel miglioramento dei test.

2.2 Area di processo 2.2 Pianificazione dei test

Lo scopo della pianificazione dei test è quello di definire un approccio ai test basato sui rischi identificati e sulla strategia di test definita, e di stabilire e mantenere piani fondati per l'esecuzione e la gestione delle attività di test.

Attenzione: la chiave del successo della pianificazione dei test sta nella riflessione iniziale ("l'attività"), non nella definizione del Test Plan associato ("il documento").

Per i cicli di vita Agile, in genere si verificano due tipi di pianificazione: la pianificazione dei rilasci e la pianificazione delle iterazioni. L'area di processo Pianificazione dei test del livello 2 del TMMi si concentra sulle attività relative ai test sia nella pianificazione dei rilasci che in quella delle iterazioni. La pianificazione del rilascio guarda al rilascio di un prodotto all'inizio di un progetto. La pianificazione del rilascio richiede un backlog di prodotto definito e può comportare il perfezionamento delle User Story più grandi in un insieme di storie più piccole. La pianificazione del rilascio fornisce la base per un approccio e un Test Plan che copra tutte le iterazioni. I piani di rilascio sono di alto livello. Dopo che la pianificazione del rilascio è stata fatta, inizia la pianificazione dell'iterazione per la prima iterazione. La pianificazione dell'iterazione guarda alla fine di una singola iterazione e si occupa del backlog dell'iterazione..

Si noti che l'area di processo Pianificazione dei test ha una serie di pratiche specifiche. Il TMMi non indica quando o come condurre queste pratiche. Non dice nemmeno che non si può pianificare in modo incrementale. L'approccio tradizionale è stato quello di consolidare il maggior numero possibile di decisioni in anticipo, in modo da poter consolidare anche i costi e le tempistiche relative. La logica di questo approccio è stata quella di stimare meglio il lavoro e di ridurre il rischio di "Scope Creep". Gli approcci agili, invece, sono generalmente improntati all'idea di ottenere un maggior valore attraverso il continuo perfezionamento del piano sulla base delle informazioni più recenti e della collaborazione continua con il cliente.

2.2.1 SG1 Eseguire la valutazione del rischio

È impossibile fare test esaustivi e bisogna sempre fare delle scelte e stabilire delle priorità. Questo obiettivo di TMMi è quindi applicabile anche ai progetti Agile. Per i progetti Agile, al momento della pianificazione del rilascio deve essere eseguita una valutazione del rischio di prodotto di alto livello basata su un documento di visione del prodotto o su un insieme di User Story di alto livello. Per ogni iterazione, nell'ambito della sessione di pianificazione dell'iterazione, deve essere eseguita una sessione di rischio prodotto più dettagliata basata sulle User Story o su altri requisiti per quell'iterazione. Il processo di valutazione del rischio di prodotto in un progetto Agile avrà un formato molto più leggero rispetto a quello applicato nei progetti tradizionali che seguono un modello di ciclo di vita sequenziale. Un esempio di tecnica di rischio di prodotto leggera da utilizzare è il Risk Poker [Van Veenendaal].

Durante la pianificazione del rilascio, i rappresentanti del business che conoscono le caratteristiche della release forniscono una panoramica di alto livello delle funzionalità da sviluppare e l'intero team, compresi i tester, contribuirà all'identificazione e alla valutazione dei rischi.

Durante la pianificazione dell'iterazione, il team Agile identifica e analizza i rischi di prodotto sulla base delle User Story da implementare nella iterazione successiva. Preferibilmente tutti i membri del team Agile ed eventualmente altri stakeholder partecipano alla sessione sui rischi di prodotto. Il risultato è un elenco prioritario di elementi di rischio

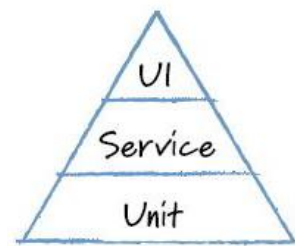
del prodotto che identifica le aree critiche da testare. Questo, a sua volta, aiuterà a determinare la quantità appropriata di sforzi di test da allocare per coprire ogni rischio con un numero sufficiente di test, e a sequenziare questi test in modo da ottimizzare l'efficacia e l'efficienza del lavoro di test da svolgere. I Task stimati sulla Task Board possono essere prioritari in parte in base al livello di rischio del prodotto ad essi associato. I Task associati a rischi più elevati dovrebbero iniziare prima e comportare un maggiore impegno di test. I Task associati a rischi più minori dovrebbero iniziare più tardi e comportare un minore impegno di test.

2.2.2 SG2 Stabilire un approccio di test

Viene definito un approccio di test per mitigare i rischi di prodotto identificati e prioritari. Per un'iterazione specifica, gli elementi e le caratteristiche da testare vengono identificati durante la pianificazione dell'iterazione. Questa attività si basa anche sui risultati della sessione sui rischi del prodotto. L'elenco prioritario degli elementi da testare si riferisce in genere alle User Story da testare in questa iterazione. Le caratteristiche si riferiscono tipicamente, tra l'altro, alle varie caratteristiche di qualità del software da testare. Durante l'iterazione possono emergere nuovi rischi di prodotto che richiedono ulteriori test. Le questioni come i rischi di nuovi prodotti che richiedono test aggiuntivi sono tipicamente discusse durante le riunioni giornaliere di standup.

L'approccio di test definito a livello di iterazione per mitigare i rischi può riguardare, ad esempio, la revisione aggiuntiva delle User Story e dei criteri di accettazione, l'impegno di test proporzionale al livello di rischio, la selezione di tecniche di test appropriate in base al livello e al tipo di rischio. L'approccio di test a livello di rilascio sarà a un livello molto più alto e si baserà sulla strategia di test definita a livello di programma o di organizzazione. Spesso un approccio di test viene tenuto o visualizzato sul wiki del team/progetto.

Un rischio importante, sempre presente nello sviluppo iterativo, è il rischio di regressione. L'approccio di test deve definire come gestire il rischio di regressione. In genere, ciò avviene attraverso la creazione di uno specifico set di test di regressione, preferibilmente automatizzato. In questo contesto è utile la piramide dell'automazione dei test [Cohn]. Essa mostra come massimizzare il valore dell'automazione dei test di regressione, partendo dai test unitari al livello più basso della piramide e passando per i test a livello di servizio. Il test dell'interfaccia utente si trova al vertice. I test unitari sono veloci e affidabili. Il livello di servizio consente di testare la logica di business a livello di API o di servizio, senza essere vincolati dall'interfaccia utente (UI). Più si sale di livello, più i test diventano lenti e fragili.



I criteri di ingresso, normalmente parte di un approccio di test definito (pratica specifica 2.3), probabilmente non sono rilevanti per lo sviluppo Agile. Nello sviluppo software Agile, il testing è parte integrante del processo del team e un'attività quasi continua. Di conseguenza, non c'è bisogno di una Check List o di un gateway specifico per determinare se il test può o non può essere avviato. Questo vale anche per un componente che passa, all'interno di un team Agile, da una fase di test (ad esempio, il test unitario) a un'altra (ad esempio, il test di accettazione).

I criteri di uscita dal test (pratica specifica 2.4) fanno parte della cosiddetta Definition of Done (DoD). È importante che la DoD contenga criteri specifici relativi ai test, ad esempio per la copertura dei test e la qualità del prodotto (difetti). L'iterazione deve portare all'implementazione dell'insieme di User Story concordate e soddisfare i criteri di uscita (di test) definiti nella DoD. In genere le storie che non soddisfano i criteri di uscita vengono inserite nel backlog e possono essere trattate nell'iterazione successiva. Naturalmente in Agile non esistono criteri di uscita per un componente che passa da una fase di test a un'altra. Non solo esiste una Definition of Done a livello di iterazione, ma spesso esiste anche una DoD a livello di release che abbraccia più iterazioni. Anche la DoD a livello di release avrà tipicamente criteri di copertura e di qualità del prodotto.

La pratica specifica 2.5 Definire i criteri di sospensione e ripresa è probabilmente non rilevante per i cicli di vita Agile. Poiché il testing è parte integrante del processo di sviluppo software Agile, non sarà ovviamente trattato come un'attività separata e indipendente dalle altre attività di iterazione. Quando ci sono problemi di blocco che possono essere considerati come minacce potenziali o reali all'avanzamento dei test, questi vengono discussi nella riunione giornaliera di standup. In questa sede il team deciderà le eventuali azioni da intraprendere per risolvere i problemi. Non saranno quindi richiesti e definiti criteri formali di sospensione e ripresa, ma le questioni relative a questo aspetto

vengono affrontate come parte della normale routine Agile. La routine Agile serve quindi come pratica alternativa per questa pratica specifica.

2.2.3 SG3 Stabilire le stime dei test

Per i team Agile, le stime dettagliate per i test vengono effettuate durante la pianificazione delle iterazioni. Le stime di alto livello (di test) vengono fatte durante la pianificazione del rilascio ed eventualmente anche durante le sessioni di rifinitura del backlog. Tutte le stime sono ovviamente effettuate come stime di team, che includono tutti gli sforzi necessari per la realizzazione di ciascuna storia. È importante assicurarsi che le attività di test siano effettivamente prese in considerazione durante le sessioni di stima. Questo può essere fatto identificando le attività di test come Task separati e stimando successivamente ogni singolo Task di test, oppure stimando ogni User Story in cui i test da fare sono esplicitati e quindi presi in considerazione. Un tester, facendo parte del team Agile, deve partecipare alle sessioni di stima. Il Poker di pianificazione o la dimensione della camicia sono tecniche di stima tipiche dello sviluppo software Agile.



Il lavoro da svolgere per l'iterazione successiva è tipicamente definito dalle User Story. Le User Story devono essere di dimensioni ridotte per essere stimabili. La stimabilità è uno dei criteri delle User Story definiti da INVEST [Wake] ed è applicabile alle User Story che devono far parte di un'iterazione. Durante la pianificazione dell'iterazione, i team Agile tipicamente identificano e definiscono i Task di test. I Task di test vengono inseriti in una task-board insieme agli altri Task di sviluppo. Questo insieme di Task è la base per la stima dell'iterazione. L'insieme dei Task definiti per l'iterazione successiva serve come una sorta di struttura di interruzione del lavoro (come si usa nei progetti sequenziali).

Al momento della pianificazione del rilascio, le User Story o le Epic saranno tipicamente definite ad alto livello e non ancora dettagliate in Task specifici. Questo naturalmente renderà le stime più difficili e meno accurate. Come detto, i progetti Agile non stabiliscono una struttura di work-breakdown come base per le stime, ma a livello di release possono beneficiare di un semplice diagramma che visualizzi il prodotto da sviluppare e quindi definisca adeguatamente lo scope del lavoro.

Sebbene un team Agile effettui le stime in modo relativamente informale, la motivazione delle stime deve essere chiara (cioè quali fattori vengono presi in considerazione). Le discussioni basate sulle motivazioni promuovono un livello più elevato di accuratezza delle stime. In genere, nei progetti Agile la stima si concentra sulle dimensioni (usando gli Story Point) o sull'impegno (usando i giorni uomo ideali come unità di stima). I costi non vengono normalmente affrontati come parte delle sessioni di stima nei progetti Agile.

Questo obiettivo specifico TMMi e le sue pratiche specifiche sono quindi pienamente applicabili, ad eccezione della pratica specifica 3.2 Definire il ciclo di vita dei test. Una delle basi dello sviluppo iterativo e agile del software è lavorare a piccoli pezzi. Pertanto, i Task che sono stati identificati sono in genere abbastanza dettagliati da servire come base per la stima (dei test). In Agile, quindi, non è necessario definire un ciclo di vita per le attività di test che serva da base aggiuntiva per la stima.

2.2.4 SG4 Sviluppare un Test Plan

Ribadisco l'osservazione che la pianificazione dei test riguarda la riflessione iniziale ("l'attività") e non la definizione del Test Plan associato ("il documento"). In Agile la maggior parte delle attività di pianificazione dei test, così come definite da questo obiettivo specifico del TMMi, saranno svolte durante la pianificazione dei rilasci e delle iterazioni. Tuttavia, il risultato di queste attività non sarà tipicamente documentato in un Test Plan, soprattutto per la pianificazione dell'iterazione, dove potrebbe essere riportato sulla task board.

Poiché il testing viene eseguito come parte integrante della pianificazione dei rilasci e delle iterazioni, le "schedulazioni" risultanti includeranno anche le attività di testing. Piuttosto che un calendario dettagliato, come quello sviluppato con i cicli di vita sequenziali, il calendario di un progetto Agile è molto più simile a un ordine di User Story

(voci del backlog) e attività che riflette le priorità di rilascio e di iterazione, ad esempio in base alla delivery del Business Value desiderato. Le mappe mentali sono spesso utilizzate come tecnica di supporto. La scheda dei Task rifletterà le priorità dell'iterazione. Non viene quindi stabilito un calendario esplicito (di test); ci si aspetta che vengano definite chiare priorità di rilascio e di iterazione per le User Story, rispettivamente per i Task da svolgere, compresi quelli di test.

A livello di progetto, la selezione del team di test fa parte della costruzione del team; la necessità di risorse di test o multi-specializzate viene identificata in anticipo. Quando i progetti cambiano o crescono, si può facilmente dimenticare la motivazione della selezione iniziale di un individuo per un determinato team/progetto, che spesso include le esigenze di competenze specifiche o l'esperienza specificamente legata al team/progetto. Questo fornisce una buona motivazione per annotare le esigenze di competenze delle persone, fornendo informazioni di supporto relative al motivo per cui sono state selezionate per il team/progetto. Una volta definiti i team Agile, il team di test è più o meno fisso. Durante la pianificazione dell'iterazione, l'identificazione delle risorse e delle competenze (aggiuntive), ad esempio per i test non funzionali, necessarie per eseguire i test in un'iterazione può essere discussa, se necessario, per garantire che il team abbia risorse, conoscenze e competenze sufficienti per eseguire i test richiesti.

Lo Scrum master deve assicurarsi che il Product Owner fornisca input al testing come richiesto, ad esempio rispondendo a domande e conversando sulle storie degli utenti. Il Product Owner deve avere una disponibilità sufficiente, che deve essere organizzata in anticipo.

Una sessione iniziale sui rischi del progetto dovrebbe far parte della pianificazione del rilascio e dell'iterazione. L'identificazione (e la gestione) di ulteriori rischi di progetto durante l'iterazione fa parte delle riunioni giornaliere di standup e viene tipicamente documentata per mezzo di un Impediment Log. È importante che anche i problemi di test siano annotati nell'Impediment Log. Gli impedimenti devono essere discussi durante la riunione giornaliera di standup, finché non vengono risolti.

Anche se non viene sviluppato e documentato un Test Plan specifico e dettagliato, la pratica specifica 4.5 Stabilire il Test Plan è ancora rilevante in un contesto Agile. Il risultato delle discussioni che si svolgono nell'ambito della pianificazione dei test è tuttavia suscettibile di essere catturato in una forma leggera, possibilmente una mappa mentale.

2.2.5 SG5 Ottenere l'impegno per il Test Plan

Nell'ambito di Agile, il processo di sviluppo e definizione di un approccio e di un Test Plan è un esercizio basato sul team, eventualmente guidato da un professionista del test (che è uno dei membri del team). La qualità del prodotto è una responsabilità del team. Pertanto, a condizione che il team segua il processo corretto, l'impegno verso l'approccio (di test) e il piano (di test) è già un risultato implicito della pianificazione del rilascio e dell'iterazione, poiché si tratta di uno sforzo di team. Questa è ovviamente un'enorme differenza rispetto al modo di lavorare in un ambiente tradizionale, dove di solito il tester prepara il Test Plan e poi deve ottenere un impegno esplicito.

Nei progetti Agile, il team Agile (compreso il Product Owner) deve comprendere e concordare l'elenco prioritario dei rischi del prodotto e le azioni di mitigazione dei test da eseguire. La comprensione e l'impegno possono essere ottenuti, ad esempio, con una breve presentazione seguita da una discussione durante la pianificazione del rilascio o dell'iterazione che spieghi al team i rischi del prodotto, l'approccio al test e le sue motivazioni.

Durante la sessione di stima (vedere SG 3 Stabilire le stime dei test) viene stimato il carico di lavoro. Le risorse (di test) per un team sono fissate dalla configurazione del team Agile. Le User Story stimate e selezionate per essere sviluppate prendono le risorse disponibili come punto di partenza (vincolo). Quindi, anche in questo caso, riconciliare i livelli di lavoro e di risorse non è un'attività significativa. La pratica specifica 3.2 Riconciliare i livelli di lavoro e di risorse è quindi una pratica tipicamente non rilevante in un contesto Agile. Le riunioni giornaliere di stand-up saranno utilizzate per affrontare qualsiasi problema di risorse durante l'iterazione e per (ri)allocare immediatamente le risorse appropriate, oppure per rimuovere un deliverable da un'iterazione per riconciliarlo nella pianificazione di iterazioni o rilasci futuri.

2.3 Area di processo 2.3 Monitoraggio e controllo dei test

Lo scopo del monitoraggio e del controllo dei test è quello di fornire una comprensione dell'avanzamento dei test e della qualità del prodotto, in modo da poter intraprendere azioni correttive appropriate quando l'avanzamento dei test si discosta significativamente dal piano e la qualità del prodotto si discosta significativamente dalle aspettative.

Seguendo il manifesto Agile e i principi che lo accompagnano, ci sono alcune cose che sono fondamentalmente diverse dal monitoraggio e dal controllo nei progetti Agile rispetto ai progetti tradizionali. Sebbene il monitoraggio e il controllo siano elementi essenziali di un progetto Agile, ciò non implica che l'obiettivo sia attenersi a un piano rigido, anzi è vero il contrario: sia il manifesto che i principi parlano di accogliere il cambiamento. Il monitoraggio e il controllo dei test potrebbero essere interpretati, in un contesto Agile, come la fornitura di best practice per aggiustare continuamente il piano per mantenerlo aggiornato, come raccomandano gli approcci Agile.

Dal punto di vista del monitoraggio e del controllo dei test, ciò significa che non siamo guidati dal piano, ma che rivediamo costantemente i nostri progressi e i risultati dei test, adattando il piano e l'approccio in modo appropriato - possono emergere nuovi rischi del prodotto. Il Test Plan, in qualsiasi modo sia stato realizzato, è un'entità viva e deve essere continuamente rivisto e aggiornato in base alle nuove informazioni o ai feedback ricevuti. I progetti agili devono anche tenere a mente il "quadro generale" e monitorare e controllare a livello superiore (release) e a livello di iterazione.

È importante notare che, essendo il testing un processo pienamente integrato nel processo complessivo del team Agile, anche il monitoraggio e il controllo dei test sono parte integrante dei meccanismi complessivi di monitoraggio e controllo del team Agile. Di conseguenza, i tester non riportano a un test manager come nei progetti tradizionali, ma al team.

Poiché l'area di processo Pianificazione dei test del livello 2 del TMMi si concentra sulla pianificazione di release e iterazioni, lo stesso vale per il monitoraggio e il controllo dei test. Ci si aspetta che il monitoraggio e il controllo dei test vengano eseguiti sia a livello di pianificazione che di rilascio.

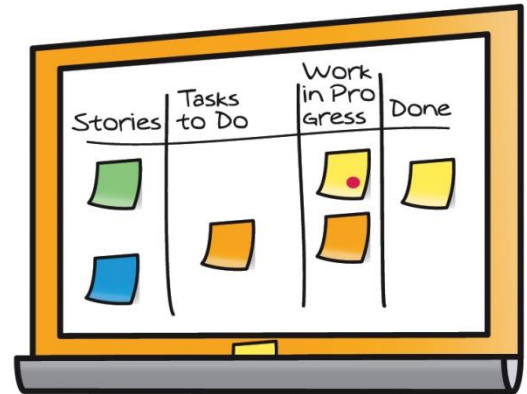
2.3.1 SG1 Monitoraggio dell'avanzamento dei test rispetto al piano

I tester nei team Agile utilizzano vari metodi per monitorare e registrare i progressi dei test, ad esempio progressione dei Task e delle Story di test sulla task board Agile e grafici di burndown. Questi possono poi essere comunicati al resto del team utilizzando mezzi come wiki dashboard ed email in stile dashboard, oltre che verbalmente durante le riunioni di stand-up. I team possono usare i grafici di burndown sia per tracciare i progressi dell'intera release sia all'interno di ogni iterazione. Un grafico di burndown mostra tipicamente i progressi rispetto alla velocità prevista e al backlog delle funzionalità da implementare (prestazioni del team). A volte, quando le risorse dell'ambiente di test sono scarse e vitali, ad esempio per i test non funzionali, si utilizzano grafici di burndown specifici per monitorare l'utilizzo delle risorse dell'ambiente di test rispetto a quelle concordate durante la pianificazione dell'iterazione. Un'altra pratica comune è quella di tenere traccia dei problemi relativi all'ambiente di test attraverso la Task Board, ad esempio utilizzando adesivi con la dicitura "ambiente di test bloccato" sulle story card o creando una colonna separata sulla lavagna dove tutte le storie bloccate dagli ambienti attendono di essere sbloccate. Si tratta di dare visibilità all'impatto di un ambiente di test che blocca i progressi.

Per fornire una rappresentazione visiva istantanea e dettagliata dello stato attuale dell'intero team, compreso lo stato dei test, i team possono utilizzare i task board Agile. Le story card, i Task di sviluppo, i Task di test e altri Task creati durante la pianificazione dell'iterazione sono catturati sulla task board. Vengono registrati sulla Task Board, spesso utilizzando schede coordinate per colore per determinare il tipo di Task. Durante l'iterazione, i progressi sono gestiti attraverso lo spostamento di questi Task sulla task board in colonne come *da fare*, *in corso d'opera* e *completato*. I team Agile possono utilizzare strumenti per mantenere le loro story card sui task board Agile, che possono automatizzare le dashboard e la panoramica dello stato. Tuttavia, alcuni team non creano Task specifici per le singole attività, ma si limitano a utilizzare la story card e ad annotarvi i commenti per i test, facendo riferimento a strumenti Agile o a wiki in cui i test possono essere documentati. I Task di test sulla scheda di attività si riferiscono tipicamente ai criteri di accettazione definiti per le User Story. Come script di automazione dei test Quando gli script di automazione dei test, i test manuali scriptati e i test esplorativi per un Task di test raggiungono lo stato di

superamento, il Task si sposta nella colonna done della task board.

La Definition of Done serve come criterio di uscita rispetto al quale si misurano i progressi. La DoD dovrebbe anche essere correlata alle attività di test e mostra tutti i criteri che devono essere soddisfatti prima che il test di una User Story possa essere definito "Fatto". Si noti che i criteri di test relativi alle attività di test costituiscono solo una parte di ciò che il team accetta di completare come Definition of Done. I criteri di definizione di "Fatto" sono tipicamente applicati a più livelli, ad esempio a livello di iterazione e di rilascio. L'intero team esamina regolarmente lo stato della task board, spesso durante le riunioni giornaliere di stand-up, per assicurarsi che i Task si stiano muovendo a un ritmo accettabile. Se qualche attività (comprese quelle di test) non si muove o si muove troppo lentamente, si avvia una discussione di gruppo in cui si analizzano i problemi che possono bloccare l'avanzamento di tali attività.



La riunione giornaliera di stand-up comprende tutti i membri del team Agile, compresi i tester. In questa riunione, i tester comunicano al resto del team il loro stato attuale e i progressi effettivi. Qualsiasi problema che possa bloccare l'avanzamento dei test viene comunicato durante le riunioni giornaliere di stand-up, in modo che l'intero team sia a conoscenza dei problemi e possa agire di conseguenza. In questo modo anche la gestione del rischio è integrata in queste riunioni quotidiane. Qualsiasi rischio del progetto, compresi quelli relativi ai test, ad esempio la mancanza di disponibilità degli ambienti di test, può essere comunicato e affrontato durante lo stand-up giornaliero. (Si noti che il monitoraggio dei rischi del prodotto fa parte del monitoraggio della qualità del prodotto e quindi viene discusso in seguito con l'obiettivo specifico SP 2 Monitoraggio della qualità del prodotto rispetto al piano e alle aspettative). Le riunioni giornaliere di stand-up per la gestione dei Task quotidiani e le pratiche del team Agile relative alla correzione del corso dei Task sono buone tecniche collaudate che soddisfano l'intento delle pratiche specifiche di TMMi nell'area di processo Monitoraggio e controllo dei test.

La revisione delle milestone in Agile avviene al completamento di un'iterazione. I risultati dei test, ad esempio rispetto alla Definition of Done, faranno parte della revisione dell'iterazione. Le dimostrazioni vengono organizzate con gli stakeholder per discutere il valore di business e la qualità del prodotto consegnato.

Gli stakeholder sono rappresentati dal Product Owner nella pianificazione dell'iterazione, nelle revisioni dell'iterazione (demo) e nelle retrospettive. Il Product Owner è coinvolto nelle discussioni sul backlog del prodotto e fornisce feedback e input per l'elaborazione delle User Story e la progettazione dei test durante l'iterazione. Gli altri stakeholder sono coinvolti alla fine di ogni iterazione durante la revisione dell'iterazione. Non è richiesto un monitoraggio specifico del coinvolgimento degli stakeholder, in quanto la loro rappresentazione da parte del Product Owner è parte integrante del metodo di lavoro Agile. Si noti che il coinvolgimento attivo del Product Owner nei progetti Agile è un fattore di successo cruciale, ma a volte è piuttosto difficile da realizzare nella pratica. In quest'ultimo caso, il problema verrà segnalato e discusso durante lo stand-up giornaliero e gestito come un rischio di progetto (vedi sopra) che influisce sull'efficacia e sull'efficienza dell'intero team.

2.3.2 SG2 Monitoraggio della qualità del prodotto rispetto al piano e alle aspettative

Per il monitoraggio della qualità del prodotto si utilizzano in gran parte gli stessi meccanismi del monitoraggio dell'avanzamento (vedi SG1 sopra). In Agile, per il monitoraggio dei rischi di prodotto l'attenzione si concentra sulla revisione dell'elenco dei rischi di prodotto nel corso di riunioni periodiche, piuttosto che sulla revisione di una documentazione dettagliata dei rischi. I rischi di prodotto appena identificati o modificati, ad esempio in seguito a test esplorativi, vengono discussi e vengono concordate le azioni di test necessarie. Lo stato dei vari rischi di prodotto viene tipicamente mostrato mediante grafici sul cruscotto.

La prassi migliore è che nessuna funzionalità sia considerata completata finché non è stata integrata e testata con successo nel sistema. In alcuni casi, vengono effettuate periodicamente iterazioni di hardening o di stabilizzazione per risolvere eventuali difetti persistenti e altre forme di debito tecnico. Per monitorare e migliorare la qualità del prodotto, i team Agile utilizzano metriche basate sui difetti, simili a quelle delle metodologie di sviluppo tradizionali,

come i tassi di superamento/fallimento dei test, i tassi di scoperta dei difetti, i difetti trovati e risolti. Il numero di difetti trovati e risolti durante un'iterazione e il numero di difetti non risolti, che potrebbero entrare a far parte del backlog dell'iterazione successiva, devono essere monitorati durante le riunioni giornaliere di stand-up. Per monitorare e migliorare la qualità complessiva del prodotto, molti team Agile utilizzano anche survey sulla soddisfazione del cliente per ricevere un feedback sulla rispondenza del prodotto alle sue aspettative.

I criteri di uscita dei test, ad esempio per la copertura dei test e la qualità del prodotto (difetti), fanno parte della Definition of Done (DoD). L'aderenza ai criteri di uscita concordati è tipicamente monitorata attraverso la task board, in base alla quale una storia può essere indicata come "fatta" solo se è conforme ai suoi criteri DoD.

La riunione giornaliera di stand-up è il meccanismo utilizzato per eseguire quasi continuamente revisioni della qualità del prodotto. La pietra miliare della revisione della qualità del prodotto in Agile è il completamento di un'iterazione. Vengono organizzate dimostrazioni con gli stakeholder per discutere il valore di business e la qualità del prodotto consegnato. La qualità del prodotto viene verificata e convalidata rispetto ai criteri di qualità DoD definiti.

Nell'ambito dell'area di processo Pianificazione dei test, le pratiche specifiche sul monitoraggio dei criteri di ingresso, sospensione e ripresa sono probabilmente non rilevanti. Per ulteriori informazioni, si rimanda all'obiettivo specifico 2 Stabilire un approccio di test dell'area di processo Pianificazione dei test, in cui è stata fornita una spiegazione del perché questi criteri sono tipicamente non rilevanti in un ambiente Agile.

2.3.3 SG3 Gestire le azioni correttive fino alla loro chiusura

I team Agile noterebbero molto rapidamente problemi come deviazioni dalle aspettative in un grafico di burn-down e/o la mancanza di progressione di task e storie (di test) sulla task board Agile. Questi e altri problemi, ad esempio quelli che bloccano l'avanzamento dei test, vengono comunicati durante le riunioni giornaliere di stand-up (vedi sopra), in modo che tutto il team sia a conoscenza dei problemi. In questo modo si può innescare una discussione di gruppo in cui si analizzano i problemi. Il risultato potrebbe essere che la baseline necessita di un aggiornamento (ad esempio, la rimozione di una o più User Story dal backlog dell'iterazione), che la Definition of Done è forse troppo rigida o che è necessario apportare modifiche al modo di lavorare. Il team deciderà le azioni correttive da intraprendere insieme al Product Owner. Nei casi in cui il backlog di iterazione viene modificato, il Product Owner presenta al team un backlog di iterazione aggiornato e gli elementi rimossi dal backlog di iterazione vengono portati all'iterazione successiva e discussi durante la pianificazione dell'iterazione.

La gestione delle azioni correttive nei progetti Agile è principalmente una responsabilità del team auto-organizzato. Il team può definire e implementare le azioni correttive appropriate, oppure segnalare qualsiasi problema come "impedimento" allo Scrum master. Lo Scrum master ha tipicamente la responsabilità di supportare il team nella gestione dei problemi fino alla loro chiusura. Gli eventi tipici per discutere e gestire le azioni correttive sono gli stand-up giornalieri e le riunioni retrospettive. Le azioni correttive concordate possono essere gestite fino alla chiusura come "attività" o "voci arretrate" attraverso il product backlog o (all'interno dell'iterazione) attraverso la task board.

2.4 Area di processo 2.4 Progettazione ed esecuzione dei test

Lo scopo di Test Design and Execution è quello di migliorare la capacità del processo di test durante la progettazione e l'esecuzione dei test, stabilendo le specifiche di progettazione dei test, utilizzando le tecniche di progettazione dei test, eseguendo un processo di esecuzione dei test strutturato e gestendo gli incidenti di test fino alla chiusura.

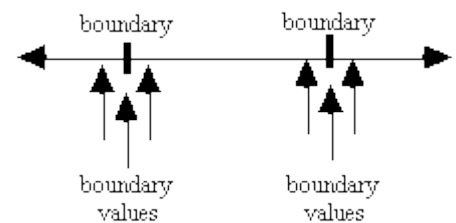
Sebbene l'obiettivo di fondo ("mitigare i rischi e testare il software") sia lo stesso sia per un progetto Agile che per uno sequenziale tradizionale, l'approccio adottato sulle modalità di test è tipicamente molto diverso. In Agile, la flessibilità e la capacità di rispondere ai cambiamenti sono punti di partenza importanti. Inoltre, l'analisi e la progettazione dei test, l'implementazione e l'esecuzione dei test non sono fasi di test successive, ma vengono eseguite in parallelo, sovrapposte e in modo iterativo. Il livello di dettaglio della documentazione di test stabilita è un'altra differenza fondamentale. In genere, nei progetti Agile si utilizzano più spesso tecniche basate sull'esperienza e sui difetti, anche se le tecniche di progettazione dei test basate sulle specifiche possono essere ancora applicabili e utilizzate. Con una maggiore enfasi sui test di unità e di integrazione, sono molto più diffuse anche le tecniche white-box, come gli statement e i decision test. Un'ultima differenza importante è il livello del rischio di regressione, che richiede un maggior numero di test di regressione ai vari livelli di test. Idealmente, i test di regressione sono

altamente automatizzati. Le differenze sono molte, ma alla fine nel contesto dell'area di processo Test Design and Execution si tratta sempre di creare test, mitigare i rischi dei prodotti, eseguire i test e trovare i difetti.

2.4.1 SG1 Eseguire l'analisi e la progettazione dei test utilizzando le tecniche di progettazione dei test.

In Agile, l'analisi e la progettazione dei test e l'esecuzione dei test sono attività che si supportano a vicenda e che di solito si svolgono in parallelo nel corso di un'iterazione. Nei progetti a ciclo di vita sequenziale, l'analisi dei test avviene attraverso la revisione da parte dei tester della base di test, ad esempio i requisiti, e la valutazione della testabilità della base di test una volta creata. Nei progetti Agile, i tester fanno parte di un team che collabora alla creazione e al perfezionamento delle User Story. Durante lo sviluppo dei requisiti vengono eseguite frequenti revisioni informali, che includono i criteri di accettazione per ogni User Story. Questi criteri sono definiti in collaborazione tra rappresentanti del business, sviluppatori e tester. In genere, il punto di vista unico del tester migliorerà la User Story identificando i dettagli mancanti e rendendoli testabili. Un tester può contribuire ponendo ai rappresentanti dell'azienda domande aperte sulla User Story e sui suoi criteri di accettazione e proponendo modi per testare la User Story. L'analisi dei test non è quindi un'attività esplicita separata, ma piuttosto un'attività implicita che i tester svolgono come parte del loro ruolo all'interno dello sviluppo collaborativo della User Story.

Sulla base dell'analisi delle User Story, vengono identificate le condizioni di test¹. Dal punto di vista del testing, la base di test viene analizzata per vedere cosa può essere testato - queste sono le condizioni di test. Le condizioni di test (talvolta indicate come situazioni di test) sono fondamentalmente un'identificazione delle "cose" che devono essere testate/coperte [Black, Van Veenendaal]. Se i criteri di accettazione definiti sono sufficientemente dettagliati e chiari, possono assumere il ruolo di condizioni di test tradizionali. Le condizioni di test vengono successivamente tradotte in test, necessari per fornire la copertura dei criteri di accettazione definiti e concordati. Spesso è anche utile eseguire l'analisi dei test a un livello più alto rispetto alle sole User Story. Ad esempio, analizzando una caratteristica o un'Epic o un insieme di storie per identificare le condizioni di test che sono più astratte di quelle a livello di User Story e che coprono anche più User Story. Le tecniche di progettazione dei test basate sulle specifiche sono tipicamente utili per derivare le condizioni di test dalle User Story e dai criteri di accettazione. Tuttavia, in Agile il più delle volte queste tecniche di progettazione dei test sono utilizzate più implicitamente che esplicitamente, in quanto i tester, in base alla loro esperienza, hanno acquisito la padronanza delle tecniche e sono in grado di utilizzarle con flessibilità nel contesto. Le condizioni di test saranno documentate in un formato leggero, contrariamente al modo di lavorare più tradizionale in cui sono documentate come parte di un documento di specifiche di progettazione del test. A volte, soprattutto nei progetti in cui si fa largo uso di test esplorativi, le condizioni di test vengono identificate solo attraverso un brainstorming e documentate come idee di test (da inserire nel Test Chart). La definizione delle condizioni di test è anche la base per stabilire la tracciabilità orizzontale (si veda più avanti) e per gestire la copertura dell'insieme di test di regressione (automatizzati), sviluppando test automatizzati per coprire specifiche condizioni di test.



Con l'applicazione del principio test-first in Agile, i test che coprono l'insieme delle condizioni di test saranno identificati (e possibilmente automatizzati) prima, o almeno in parallelo, allo sviluppo del codice. Per i test unitari automatizzati si può prendere in considerazione un approccio come il Test-Driven Development. Per livelli di test più

¹ Il termine condizione di test è utilizzato in questo paragrafo sulla base delle pratiche TMMi, sebbene non sia un termine comune di Agile. Il TMMi definisce una condizione di test come "Un elemento o un evento di un componente o di un sistema che potrebbe essere verificato da uno o più casi di test, ad esempio una funzione, una transazione, una caratteristica, un attributo di qualità o un elemento strutturale". Non si intende qui imporre un ulteriore livello di documentazione di test a un team Agile. Tuttavia, è importante che il team comprenda chiaramente quali elementi (o eventi) devono essere testati. In alcuni progetti ci sono molti test, ma senza una comprensione di ciò che realmente coprono e devono coprire. In un contesto Agile, l'intento delle pratiche TMMi sulle condizioni di test è tipicamente raggiunto con pratiche alternative come i criteri di accettazione (purché specificati al livello di dettaglio richiesto) e i Test Chart nei test esplorativi.

elevati, Behavior-Driven Development (BDD) e Acceptance Test-Driven Development (ATDD) sono approcci di sviluppo Agile molto diffusi che supportano anche i test. Sia il BDD che l'ATDD sono anche molto legati all'automazione dei test.

Per la maggior parte dei test manuali, i test vengono identificati/raffinati man mano che il team procede con l'esecuzione dei test. Il più delle volte i test non sono documentati in modo così dettagliato come nei progetti tradizionali, ma piuttosto in un formato di idee di test utilizzando test esplorativi. Per le aree complesse e critiche, un approccio più tradizionale alla progettazione dei test e allo sviluppo dei casi di test utilizzando tecniche formali di progettazione dei test può essere il modo migliore per coprire i rischi. Tuttavia, anche con questo approccio la quantità di documentazione sarà limitata rispetto ai test in un ambiente tradizionale di ciclo di vita sequenziale. La priorità dei test segue la priorità della User Story che stanno coprendo. La prioritizzazione delle User Story si basa sul valore di business; la priorità più alta si riferisce al valore di business più elevato. È importante che i test siano stabiliti per coprire i rischi funzionali e non funzionali, ma soprattutto in Agile anche per coprire il rischio di regressione.

Vengono identificati i dati di test specifici necessari per supportare le condizioni di test e l'esecuzione dei test. Tuttavia, in Agile, a differenza degli ambienti tradizionali, i dati di test necessari non sono tipicamente specificati come parte di un documento di specifiche di test. Piuttosto, se sono disponibili gli strumenti e/o le funzionalità necessarie, i dati di test vengono creati immediatamente per consentire un inizio quasi immediato dell'esecuzione dei test manuali. Tuttavia, per i test automatizzati, i dati devono essere specificati in anticipo.

È necessario stabilire e mantenere la tracciabilità tra i requisiti, le condizioni di test e i test. I team devono accertarsi di aver coperto le varie User Story e i criteri di accettazione nell'ambito dei test. Anche se non è necessario uno strumento formale per la gestione dei requisiti, il team ha bisogno di un supporto per organizzare e gestire i requisiti in modo da poter assegnare degli identificatori a ciascun requisito, in modo che questi possano essere utilizzati per garantire che i test siano completati secondo i criteri concordati. In molte organizzazioni Agile, le User Story sono la base per lo sviluppo di una serie di criteri di accettazione e quindi di test. Una volta stabiliti questi test, spesso i test stessi diventano i requisiti dettagliati concordati. L'utilizzo di questo approccio può essere sufficiente per raggiungere l'obiettivo della pratica specifica del TMMi sulla gestione dei requisiti relativa alla tracciabilità orizzontale.

2.4.2 SG2 Esecuzione del test di implementazione

L'implementazione dei test consiste nel predisporre tutto ciò che è necessario per iniziare l'esecuzione dei test. In genere, lo sviluppo della documentazione di test, ad esempio le procedure di test, per supportare l'esecuzione dei test è ridotto al minimo. Piuttosto, vengono sviluppati script di test automatizzati (di regressione) e viene data loro priorità. Inoltre, la preparazione dei test di regressione inizia il prima possibile, parallelamente alle altre attività di test.

In un ambiente Agile le procedure di test dettagliate non sono una pratica comune. Lavorando in un team competente, i test saranno probabilmente documentati a un livello di astrazione molto più alto. Ciò sarà sufficiente per coloro che eseguono i test, poiché ci si aspetta che abbiano il livello di conoscenza tecnica e di dominio richiesto per eseguire i test. Chi esegue i test lavora all'interno del team, quindi avrà una migliore comprensione di cosa e come è stato codificato e di come la funzionalità sia adatta allo scopo. Poiché il livello di cambiamento all'interno e tra le iterazioni è tipicamente elevato, lo sviluppo di procedure di test dettagliate creerebbe anche molto lavoro di manutenzione. In genere, molti test vengono eseguiti utilizzando test esplorativi. Nei test esplorativi non vengono sviluppate procedure di test dettagliate, ma piuttosto Test Chart di una pagina di alto livello che descrivono idee di test e indicazioni per il tester. Tuttavia, i dati specifici necessari per eseguire i test devono essere creati in anticipo.

Molti team Agile utilizzano i test automatizzati. Gli approcci tipici utilizzati sono Test-Driven Development (TDD), Behavior-Driven Development (BDD) e Acceptance Test-Driven Development (ATDD). Utilizzando uno o più di questi approcci, gli script di test automatizzati vengono creati come parte dell'attività di implementazione dei test.

La Continuous Integration è una pratica fondamentale per i progetti Agile. Il test è un'attività parallela e completamente integrata che deve essere eseguita dal team. Non è un'attività o una fase separata e indipendente. La pratica specifica 2.3 Specificare la procedura di test di assunzione diventa quindi irrilevante in un team Agile. Non è necessario, anzi non ha senso, che venga specificato ed eseguito uno specifico test di ingresso formale. Tuttavia, nel caso in cui alcuni test, ad esempio il test di integrazione del sistema o il test di integrazione hardware/software, vengano eseguiti al di fuori dell'ambito del team Agile, si può specificare un test di ingresso per valutare la qualità dell'output del team Agile e determinare se il prodotto consegnato è pronto per il loro test.

In ambienti Agile non viene creato un programma specifico per l'esecuzione dei test e quindi la pratica specifica 2.4 Sviluppare un programma di esecuzione dei test è probabilmente irrilevante. Esiste un elevato livello di flessibilità per quanto riguarda l'ordine di esecuzione dei test nel corso di un'iterazione, anche se questo sarà guidato dalla priorità delle varie User Story. I vari test da eseguire saranno identificati durante la pianificazione dell'iterazione e saranno gestiti come attività di test attraverso il task board Agile.

2.4.3 SG3 Esecuzione del test

Gli elementi software sviluppati nell'iterazione devono ovviamente essere testati. Seguendo il manifesto Agile, come per gli altri obiettivi di quest'area di processo, ciò avverrà in genere con un'intensità di documentazione molto minore rispetto a un progetto sequenziale tradizionale. Spesso non vengono prodotte procedure di test dettagliate e Test Log. I test, soprattutto quelli di regressione, dovrebbero essere automatizzati per quanto possibile e fattibile.

In un progetto Agile, il software viene consegnato, come minimo, su base giornaliera e il collaudo è parte integrante del processo di sviluppo Agile, in cui il lavoro viene svolto in collaborazione reciproca per fornire un prodotto di qualità. Non esistono team di sviluppo e di test dedicati e separati. Pertanto, uno specifico test di ingresso, come definito nella pratica specifica 3.1 Eseguire il test di ingresso, probabilmente non è rilevante in un ambiente Agile puro. Tuttavia, quando alcuni test vengono eseguiti al di fuori del team Agile (come spesso accade), un test di ingresso verrà tipicamente eseguito dal team di test responsabile dei test al di fuori del team Agile. Tenete presente che ci sono attività che devono essere completate prima che inizi il test di accettazione su una specifica User Story all'interno del team Agile. Queste attività fanno tipicamente parte della Definition of Done, ad esempio il completamento e il superamento di un numero concordato di test unitari.

L'esecuzione dei test avviene in linea con le priorità definite durante la pianificazione dell'iterazione. Alcuni test possono essere eseguiti utilizzando una procedura di test documentata, ma in genere molti test vengono eseguiti utilizzando come struttura i test esplorativi e quelli basati sulle sessioni. Nei test esplorativi la progettazione e l'esecuzione dei test avvengono contemporaneamente, sotto la guida di un documento di test predisposto. Un documento di test fornisce gli obiettivi e le condizioni di test da coprire durante una sessione di test a tempo determinato. Durante i test esplorativi, i risultati dei test più recenti guidano i test successivi.

Con lo sviluppo iterativo aumenta la necessità di organizzare e strutturare i test di regressione. A volte vengono eseguiti manualmente, ma di preferenza si utilizzano test di regressione automatizzati e strumenti di supporto. I test di regressione, soprattutto quelli di unità e di integrazione, fanno spesso parte di un processo di Continuous Integration. La Continuous Integration è una pratica importante nell'ambito dello sviluppo software Agile. Si tratta fondamentalmente di un processo di compilazione e test automatizzato che si svolge almeno su base giornaliera e che rileva tempestivamente e rapidamente. La Continuous Integration consente di eseguire regolarmente test di regressione automatizzati e di inviare un rapido feedback al team sulla qualità del codice e sulla copertura del codice raggiunta.

Gli incidenti riscontrati durante i test possono essere registrati e segnalati. In genere, nei progetti Agile si discute se tutti gli incidenti riscontrati debbano essere registrati. Soprattutto nel caso in cui il team sia co-localizzato, i tester devono parlare con gli sviluppatori e gli incidenti/difetti che possono essere risolti immediatamente e inseriti nella build successiva potrebbero non essere registrati, ma solo risolti. In Agile, in genere, non tutti gli incidenti riscontrati vengono registrati e gestiti di conseguenza. Come detto, molti vengono sollevati in modo informale da un membro del team (che può essere un tester) e vengono risolti immediatamente dallo sviluppatore che ha inserito il difetto. Si perdono alcuni dati, ma si riduce l'overhead. Alcuni team registrano solo gli incidenti che sfuggono alle iterazioni, altri li registrano se non possono essere risolti oggi, altri ancora registrano solo gli incidenti ad alta priorità. Se non tutti gli incidenti rilevati vengono registrati, devono essere disponibili criteri per determinare quali incidenti devono essere registrati e quali no. Ricordate che l'intento è quello di risolvere correttamente il difetto, non di registrare un incidente. Un esempio di tale criterio: "Se un difetto può essere risolto prima del successivo standup giornaliero e della compilazione giornaliera, non è necessario documentarlo. Qualsiasi difetto sollevato che non possa essere chiuso entro un giorno, o che riguardi uno stakeholder esterno al team, deve essere documentato e catturato".

A volte gli incidenti vengono registrati sulla task board di Agile, come un adesivo su un'attività o come un'attività separata, visualizzando che stanno bloccando il completamento di una storia e delle sue attività. È accettabile utilizzare dei post-it su una task board per gestire i difetti e questo è spesso il metodo scelto dai team co-localizzati.

Il team Agile lavorerà successivamente sugli incidenti e li gestirà fino alla loro chiusura (si veda l'obiettivo specifico 4 di quest'area di processo). Alcuni scelgono di registrare e documentare gli incidenti rilevati in uno strumento, ad esempio uno strumento di gestione dei difetti o di tracciamento dei difetti. Tale strumento dovrebbe essere utilizzato nel modo più snello possibile e non costringere a presentare elementi dei rapporti sugli incidenti di test che non hanno alcun valore aggiunto o ne hanno troppo poco.

È considerata una buona pratica anche nei team Agile registrare i dati durante l'esecuzione dei test per determinare se gli elementi testati soddisfano i criteri di accettazione definiti e possono effettivamente essere etichettati come "fatti". Questo vale anche quando si applicano tecniche basate sull'esperienza, come i test esplorativi. Esempi di informazioni che può essere utile documentare sono la copertura del test (quanto è stato coperto e quanto rimane da testare), le osservazioni durante il test, ad esempio se il sistema e la User Story sotto test sembrano essere stabili, l'elenco dei rischi (quali rischi sono stati coperti e quali rimangono tra i più importanti), i difetti riscontrati e altri problemi e possibili domande aperte.

Le informazioni registrate devono essere catturate e/o riassunte in qualche forma di strumento di gestione dello stato (per esempio, strumenti di gestione dei test, strumenti di gestione delle attività, task board), in modo da rendere facile per il team e gli stakeholder comprendere lo stato attuale di tutti i test eseguiti. Naturalmente, il team non ha bisogno di conoscere solo lo stato dei Task di test isolati, ma anche lo stato generale della User Story. Nulla è completo dal punto di vista della storia se non sono state completate tutte le attività correlate, non solo i test.

2.4.4 SG4 Gestire gli incidenti di test fino alla loro chiusura

Come già detto, in ambienti Agile spesso non tutti gli incidenti rilevati vengono registrati. Questo obiettivo specifico si applica solo agli incidenti che vengono registrati e che quindi devono essere gestiti fino alla chiusura. In linea di principio, la gestione degli incidenti in Agile è semplice. Nel caso in cui l'incidente venga registrato sulla task board di Agile, come adesivo su un task o come task separato, viene visualizzato come un difetto che blocca il completamento di una storia e dei relativi task. Il metodo di lavoro Agile standard sarà applicato come per qualsiasi altro impedimento o attività che blocca il progresso. Verrà discusso durante le riunioni di stand-up e assegnato per essere risolto dal team, dopodiché sparirà dalla task board.

Nel caso in cui si decida di rimandare gli incidenti riscontrati a un'altra iterazione, essi diventano semplicemente un'altra opzione (o modifica) desiderata per il prodotto. Aggiungeteli al backlog e assegnate la priorità di conseguenza. Quando la priorità è sufficientemente alta, saranno presi in carico dal team nell'iterazione successiva. Si noti che la priorità è tipicamente definita dal business, ma a volte è necessario dare una certa priorità agli incidenti che riguardano il cosiddetto debito tecnico. Alcuni team hanno iterazioni dedicate alla pulizia degli incidenti/difetti legati al debito tecnico. Tuttavia, questa non è una pratica standard raccomandata da Agile, poiché il debito deve essere ripagato regolarmente.

Come esempio di come gli incidenti possono essere gestiti in un ambiente Agile, immaginiamo che il team sia nel bel mezzo di un'iterazione. Mentre sta lavorando su una delle User Story, il team si accorge di non riuscire a portarla a "termine", perché uno dei criteri di accettazione non è soddisfatto, ad esempio a causa di un difetto introdotto durante lo sviluppo. I team agili tendono a risolvere questo tipo di difetti il prima possibile nell'iterazione corrente, altrimenti continuerebbero a portare avanti il lavoro non fatto (equivalente a un Lean Inventory). Questo tipo di incidenti viene visualizzato e gestito attraverso la task board.

In secondo luogo, supponiamo che tutte le storie dell'iterazione vengano completate secondo la Definition of Done entro la fine dell'iterazione. Se però durante la riunione di revisione, mentre si dimostra una storia, c'è qualcosa che non si comporta come il cliente vorrebbe. Considerate che il comportamento inatteso non è stato discusso con l'azienda prima dell'inizio dell'iterazione. In questo caso, la funzionalità del software prodotto funziona e ciò che probabilmente occorre fare è tradurre l'incidente in una storia aggiuntiva e aggiungerla al backlog del prodotto.

È buona norma monitorare il numero di difetti trovati e risolti durante un'iterazione, nonché il numero di difetti non risolti che potrebbero entrare a far parte del backlog per l'iterazione successiva durante la riunione giornaliera (stand-up) e anche nelle riunioni retrospettive.

In un ambiente con un ciclo di vita sequenziale tradizionale, spesso esiste uno specifico Configuration Control Board (CCB) che si riunisce, esamina e decide la disposizione dell'incidente. In Agile, questo ruolo è assunto dal team con

maggiori poteri. Naturalmente il rappresentante del business, ad esempio il Product Owner, ha un ruolo importante da svolgere quando si tratta di decidere la disposizione di un incidente. La gestione degli incidenti è una spesa eccessiva, che sottrae attenzione e impegno ad altre attività. L'overhead aumenta quanto più cerchiamo di formalizzare un processo per elencarli, qualificarli, ordinarli e infine pianificarli. Un processo di gestione degli incidenti così complesso è un vero e proprio spreco. È meglio delineare alcuni principi di base su come gli incidenti vengono gestiti dai team Agile e agire di conseguenza.

2.5 Area di processo 2.5 Ambiente di test

Lo scopo dell'ambiente di test è stabilire e mantenere un ambiente adeguato, compresi i dati di test, in cui sia possibile eseguire i test in modo gestibile e ripetibile.

Con l'area di processo Ambiente di test viene creato e mantenuto un ambiente di test adeguato, compresi i dati di test generici, in cui è possibile eseguire i test in modo gestibile e ripetibile. Naturalmente anche in un progetto di sviluppo software che utilizza il ciclo di vita Agile, un ambiente di test adeguato è indispensabile. A causa dei brevi tempi di ciclo di un'iterazione, l'ambiente di test deve essere altamente stabile e disponibile. I problemi nell'ambiente di test avranno sempre un impatto immediato sui progressi e sui risultati dell'iterazione. La gestione corretta della configurazione e delle modifiche all'ambiente di test e ai dati di test è quindi di estrema importanza.

Gli ambienti di test sono disponibili in molti modi, a seconda, tra l'altro, del sistema in fase di test, ma in genere le specifiche, l'implementazione e la gestione dell'ambiente di test vengono effettuate in parallelo al progetto di sviluppo software Agile. L'implementazione di un ambiente di test spesso richiede molto tempo e può essere molto complessa, rendendo quasi impossibile svolgere questo Task nel tempo limitato di un'iterazione [Van der Aalst e Davis]. Inoltre, coinvolge diversi stakeholder e ingegneri che tipicamente compongono un team Agile. Le metodologie Agile sono state sviluppate per supportare lo sviluppo del software, non per lo sviluppo e la gestione dell'ambiente di test o dell'infrastruttura, ma sono meno adatte e il modo in cui l'ambiente di test o lo sviluppo e la gestione dell'infrastruttura vengono eseguiti rimane in gran parte lo stesso di un ambiente tradizionale, anche se ci possono essere alcune modifiche al processo quando si pratica Agile.

2.5.1 SG1 Sviluppare i requisiti dell'ambiente di test

La Specifica dei requisiti dell'ambiente di test viene eseguita nelle prime fasi del progetto. Le specifiche dei requisiti vengono riviste per garantirne la correttezza, l'adeguatezza, la fattibilità e la rappresentazione accurata di un ambiente operativo "reale". Una Specifica precoce dei requisiti ha il vantaggio di fornire più tempo per acquisire e/o sviluppare l'ambiente di test e i componenti necessari, come simulatori, stub o driver.

In alcuni progetti di sviluppo Agile viene applicata una cosiddetta iterazione iniziale (iterazione 0), in cui vengono elicitati e specificati i requisiti dell'ambiente di test.

2.5.2 SG2 Eseguire l'implementazione dell'ambiente di test

A volte (parte) dell'implementazione avviene nell'iterazione iniziale in aggiunta ad altre attività, ad esempio la formazione, la valutazione del rischio del prodotto ad alto livello e la Definition of Done. In questo modo si mira a far sì che anche le prime iterazioni di sviluppo del software abbiano già un ambiente di test (parzialmente) funzionante. Le azioni o le questioni tecniche ambientali possono far parte del backlog del prodotto.

La definizione dell'ambiente di test (compresi i dati di test generici) potrebbe, come in un ciclo di vita sequenziale, iniziare con un piano completo, ma spesso è meglio iniziare con l'implementazione il prima possibile e avere una versione iniziale dell'ambiente di test disponibile all'inizio della prima iterazione.

In effetti è difficile essere generici sugli ambienti di test. In alcuni ambiti gli stub e i driver sono stati rapidamente sostituiti dalla virtualizzazione dei servizi, che in genere consente un'implementazione dell'ambiente di test molto più



rapida. In alcuni progetti Agile il provisioning degli ambienti di test è altamente automatizzato. Gli ambienti di test possono essere messi in funzione in pochi minuti o ore, anziché in giorni. Oggi si fa anche un uso molto più frequente della virtualizzazione di macchine, server e servizi. Anche il cloud può aiutare a fornire gli ambienti richiesti, accelerando ancora una volta l'accesso.

2.5.3 SG3 Gestire e controllare gli ambienti di test

La gestione e il controllo dell'ambiente di test si svolgono durante tutto il progetto Agile. A volte il team Agile deve impostare e mantenere l'ambiente di test da solo, ma più spesso questo viene fatto da un'unità separata al di fuori del team Agile. Quando le attività sono svolte dal team Agile stesso, ciò implica ovviamente che il team abbia conoscenze tecniche sufficienti per poterle svolgere. Significa anche che le attività diventano parte del processo Agile, ad esempio devono essere affrontate in una sessione di pianificazione, inserite nella task board e possibilmente discusse durante le riunioni giornaliere di stand-up in caso di problemi.

Come conclusione principale per l'area di processo Ambiente di test, gli obiettivi e le pratiche specifiche sono ancora tutti applicabili e non cambiano nella sostanza. Ciò che cambia è la loro collocazione temporale nel ciclo di vita.

3 TMMi Livello 3 Definito

3.1 Area di processo 3.1 Organizzazione del test

Lo scopo dell'area di processo Organizzazione dei test è identificare e organizzare un gruppo di persone altamente qualificate responsabili dei test. Oltre ai test, il gruppo di test gestisce anche i miglioramenti al processo di test e agli asset del processo di test dell'organizzazione, sulla base di una comprensione approfondita dei punti di forza e di debolezza dell'attuale processo di test e degli asset del processo di test dell'organizzazione.

3.1.1 SG 1 Stabilire un'organizzazione del test

L'organizzazione dei test è un'area di processo spesso fraintesa. Molti la leggono come se il TMMi richiedesse un gruppo di test indipendente o addirittura un dipartimento che esegua test indipendenti. Per quanto questa sia una possibilità, esistono anche altri modelli organizzativi che soddisfano i requisiti del TMMi. Oltre all'organizzazione di test indipendente che prepara ed esegue i test, anche il centro di competenza per i test è un'opzione. I Task e le responsabilità tipici di un cosiddetto centro di competenza per i test sono la creazione, la gestione e il miglioramento dei processi di test e la fornitura di risorse ai progetti con esperienza, conoscenze e competenze in materia di test. Un centro di competenza di test ha di solito la responsabilità delle metodologie e dei processi di test. Si noti che un tester su base giornaliera è tipicamente parte del team Agile, ma oltre a ciò è comune che un tester faccia parte di un'organizzazione di test, come il centro di competenza di test o la corporazione di test.

Un'organizzazione di test richiede una leadership nelle aree che riguardano i problemi di test e di qualità. I membri del team di questo gruppo sono chiamati specialisti di test. L'opzione del centro di competenza per il test si adatta bene ad Agile e garantisce che il test sia mantenuto come disciplina e trattato seriamente come tale. Naturalmente ci sono leader di pensiero Agile che non sono favorevoli al mantenimento di discipline speciali. Sebbene questa possa essere l'opzione preferita da alcuni, la realizzazione di TMMi richiede che il testing sia trattato e mantenuto come una disciplina. Inutile dire che questo dovrebbe essere fatto solo se ha senso dal punto di vista del business e quindi quando questo non ha senso il TMMi non è probabilmente il percorso di miglioramento preferito.

In Agile, l'organizzazione dei test può assumere la forma di una squadra (guild) finalizzata al test. Una squadra di test è tipicamente un gruppo informale e autogestito di tester che fanno parte di diversi team Agile. Le attività importanti di una squadra di test possono essere la condivisione delle conoscenze, l'apprendimento di gruppo, l'osservazione delle tendenze, ecc. L'adesione a una squadra avviene su base volontaria. In pratica, una squadra di test si riunisce spesso con cadenza bisettimanale. A una squadra di test possono essere assegnati anche Task più formali, come la definizione di percorsi professionali di test, l'organizzazione della formazione e la determinazione, la pianificazione e l'implementazione di miglioramenti del processo di test. È importante garantire che la squadra di test rimanga in gran parte informale e autogestita. Un vincolo importante per il successo di una squadra di test è che ai suoi membri venga assegnato del tempo da dedicare alle varie attività della squadra di test da svolgere.

I tester indipendenti sono spesso più efficaci nel trovare i difetti. Bilanciare l'indipendenza con Agile è una sfida. Alcuni team Agile mantengono team di test completamente indipendenti e separati e assegnano i tester su richiesta durante gli ultimi giorni di ogni sprint. Questo può preservare l'indipendenza e i tester possono fornire una valutazione obiettiva e imparziale del software. Tuttavia, la pressione del tempo, la mancanza di comprensione delle nuove funzionalità del prodotto e i problemi di relazione con gli stakeholder aziendali e gli sviluppatori spesso causano problemi con questo approccio. Un'altra opzione è quella di avere un'organizzazione di test indipendente e separata, in cui i tester vengono assegnati ai team Agile su base a lungo termine, all'inizio del progetto, consentendo loro di mantenere la propria indipendenza e al contempo di acquisire una buona comprensione del prodotto e solide relazioni con gli altri membri del team. Inoltre, l'organizzazione di test indipendente può avere tester specializzati al di fuori dei team Agile per lavorare su attività a lungo termine e/o indipendenti dall'iterazione, come lo sviluppo di strumenti di test automatizzati, l'esecuzione di test non funzionali, la creazione e il supporto di ambienti e dati di test e l'esecuzione di livelli di test che potrebbero non adattarsi bene a uno sprint (ad esempio, test di integrazione del sistema). Si noti che esiste anche la possibilità di praticare il testing indipendente all'interno di un team Agile. Finché qualcun altro, oltre al creatore delle risorse o del codice, testa il prodotto, si tratta ancora di indipendenza. Pertanto, gli sviluppatori possono testare il codice degli altri tramite revisioni del codice e/o test unitari, oppure gli analisti aziendali possono contribuire ai test dal punto di vista dell'utente, ad esempio utilizzando diverse persona. Infine, ma

non meno importante, l'indipendenza può essere organizzata attraverso una struttura organizzativa e di responsabilità, ma probabilmente è soprattutto avere la giusta mentalità critica.

3.1.2 SG 2 Stabilire le funzioni di test per gli specialisti del test

Il testing è considerato una professione e una disciplina di valore. Il team Agile ha bisogno di conoscenze e competenze in materia di test. Lo specialista di test le fornirà e affiancherà gli altri membri del team nello svolgimento delle attività di test. Le funzioni di test e i percorsi di carriera sono definiti e supportati da un programma di formazione. Le conoscenze e le competenze richieste a un tipico tester Agile sono diverse da quelle di un'organizzazione tradizionale. Il tester avrà ovviamente ancora bisogno di conoscenze e competenze "tradizionali" in materia di test, ma anche di conoscenze e competenze per l'automazione dei test e per essere in grado di svolgere Task al di fuori del testing, ad esempio lo scripting o l'ingegneria dei requisiti. Poiché un tester Agile lavora in team, le soft skills sono altrettanto importanti delle crescenti competenze tecniche. Il tester Agile è un cosiddetto tester a T, che si riflette nella descrizione delle funzioni di test. L'organizzazione di test è composta da persone che hanno le competenze e la motivazione per essere buoni tester in un contesto Agile. Essi sono assegnati a una specifica funzione di test. Soprattutto in un contesto Agile è importante, dato che l'intero team svolgerà attività di test, elaborare altre descrizioni di funzioni non specifiche di test con le aspettative relative alle attività di test, ai ruoli e alle conoscenze e competenze di test richieste.



3.1.3 SG 3 Stabilire i percorsi di carriera per il test

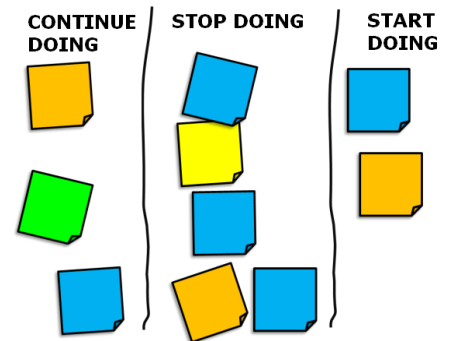
I percorsi di carriera dei test sono stabiliti per consentire ai professionisti del test di migliorare le proprie conoscenze, competenze, status e ricompense, e quindi di avanzare nella carriera. Sulla base dei percorsi di carriera definiti, vengono sviluppati e mantenuti piani di sviluppo della carriera per ogni membro dell'organizzazione di test. Questo obiettivo specifico e le pratiche specifiche che lo accompagnano sono pienamente applicabili anche in un contesto Agile. Con l'enfasi di Agile sull'aspetto delle persone, questo è chiaramente un obiettivo specifico importante. Il team Agile ha bisogno di tester che comprendano il loro difficile lavoro, che siano in grado di allenare gli altri membri del team e che siano motivati. Molti di questi aspetti vengono affrontati con i percorsi di carriera dei tester.

Tuttavia, l'effettiva definizione dei percorsi di carriera dei collaudatori sarà tipicamente diversa in un'organizzazione Agile rispetto a quella di un'organizzazione che segue un ciclo di vita sequenziale. Mentre in un'organizzazione tradizionale i percorsi di carriera dei test sono spesso incentrati sulla crescita verticale (da tester a test manager), i percorsi di carriera dei test in un'organizzazione Agile tendono a concentrarsi su una crescita orizzontale (da tester junior a tester senior e forse a test coach). In genere c'è meno bisogno di varie funzioni e ruoli gerarchici, come la gestione dei test, e una maggiore necessità di crescere lungo i ruoli di Product Owner o Scrum master. Si noti che molti dei Task tradizionali di gestione dei test sono ancora rilevanti, ma vengono assunti da altri (ad esempio, Scrum master, team Agile).

3.1.4 SG 4 Determinare, pianificare e implementare i miglioramenti del processo di test

Il processo di miglioramento descritto da questo specifico obiettivo TMMi e le sue pratiche specifiche sono ampiamente coperti dalla riunione retrospettiva eseguita da un team Agile. La retrospettiva dell'iterazione è un'opportunità per il team Agile di esaminare se stesso e definire le azioni di miglioramento da attuare durante l'iterazione successiva. Una retrospettiva avviene tipicamente dopo la revisione dell'iterazione e prima della pianificazione dell'iterazione successiva.

Durante la retrospettiva, il team discute di ciò che è andato bene nell'iterazione, di ciò che potrebbe essere migliorato e di ciò che si impegnerà a migliorare nell'iterazione successiva. Durante ogni retrospettiva, il team Agile identifica e pianifica i modi per aumentare la qualità del prodotto, ad esempio migliorando i processi di lavoro o adattando la Definition of Done, se appropriato. Le retrospettive possono portare a decisioni di miglioramento relative ai test, incentrate sull'efficacia dei test, sulla produttività dei test e sulla qualità dei test. Possono anche riguardare la testabilità delle applicazioni, delle storie degli utenti, delle funzionalità o delle interfacce del sistema.



L'analisi delle cause principali dei difetti può guidare i miglioramenti dei test e dello sviluppo. I tester svolgono un ruolo importante nelle retrospettive. Fanno parte del team e portano la loro prospettiva unica. Tutti i membri del team, tester e non, possono fornire contributi sulle attività di test e non.

Alla fine della retrospettiva, il team Agile dovrebbe aver identificato i miglioramenti (test) che implementerà nella iterazione successiva. L'implementazione di questi miglioramenti nell'iterazione successiva è l'adattamento all'ispezione del team Agile stesso. Il team vota i miglioramenti più importanti per l'iterazione successiva, quindi i miglioramenti (di test) possono rimanere nel backlog per più di un'iterazione, ma alla fine saranno implementati. È importante che il team accetti solo i miglioramenti che possono essere inseriti nell'iterazione successiva, quindi probabilmente un massimo di due o tre per iterazione. I tester devono assumersi la responsabilità dei miglioramenti dei test per garantire che vengano implementati all'interno del team. La retrospettiva è un meccanismo importante che consente a un team di evolvere e migliorare continuamente durante la vita di un progetto.

Poiché l'ambito è spesso limitato al ciclo o all'iterazione precedente, vengono apportati miglioramenti piccoli ma frequenti che si concentrano principalmente sulla risoluzione di problemi specifici del progetto. Spesso questi miglioramenti non si concentrano sull'apprendimento trasversale al progetto e sull'istituzionalizzazione dei miglioramenti. Guardando a come viene organizzato e gestito il miglioramento del processo di (test), è probabile che ci si concentri meno su un gruppo di processo di (test) a livello organizzativo e si ponga maggiormente l'accento sull'autogestione dei team all'interno del progetto. Anche se questo non è un male, è importante creare un meccanismo che permetta di condividere con il resto dell'organizzazione i miglioramenti locali dei test che hanno successo e di affrontare le aree di miglioramento che vanno oltre il potere del team Agile. Una possibile soluzione potrebbe essere quella di far partecipare un rappresentante dell'organizzazione di test alle riunioni retrospettive locali, per garantire che i miglioramenti siano condivisi e istituzionalizzati in tutta l'organizzazione, quando opportuno. Il miglioratore del processo di test può anche affrontare questioni più ampie o fondamentali che riguardano il test in tutta l'organizzazione.

3.1.5 SG 5 Distribuire il processo di test organizzativo e incorporare le lezioni apprese

Anche per un progetto Agile è importante avere accesso e riutilizzare il processo di test standard dell'organizzazione e gli asset del processo di test per quanto necessario e a valore aggiunto. L'organizzazione di test si assicurerà che questi vengano distribuiti in tutti i progetti. (Fare riferimento all'area di processo Ciclo di vita del test e integrazione per ulteriori informazioni sul processo di test standard dell'organizzazione e sugli asset del processo di test e su come questo si riferisce ai progetti Agile).

L'implementazione del processo di test standard dell'organizzazione e l'uso degli asset del processo di test nei progetti Agile possono essere monitorati dagli stessi team auto-organizzati, ad esempio affrontando questo aspetto nelle retrospettive. Se a questi incontri è presente un rappresentante dell'organizzazione di test, ad esempio un miglioratore del processo di test, questa persona può tenere traccia dello stato di implementazione e affrontare le

questioni trasversali ai progetti. Questa persona può anche riportare all'organizzazione di test le preziose lezioni apprese e i miglioramenti apportati ai test dal team Agile. Le lezioni apprese e i miglioramenti dei test possono essere successivamente presentati come proposte di miglioramento del processo di test e gestiti di conseguenza. Le lezioni apprese dai test con il team Agile vengono quindi, se opportuno, incorporate nel processo (Agile) standard dell'organizzazione e negli asset del processo di test.

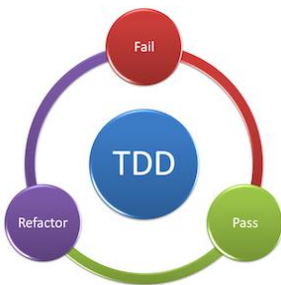
3.2 Area di processo 3.2 Programma di formazione per il test

Lo scopo dell'area di processo Programma di formazione per i test è quello di sviluppare un programma di formazione che faciliti lo sviluppo delle conoscenze e delle competenze delle persone in modo che i Task e i ruoli dei test possano essere svolti in modo efficace ed efficiente.

L'obiettivo principale del programma di formazione sui test è fornire la formazione necessaria (sui test) ai tester e agli altri membri del team. Come conseguenza dell'affermazione del manifesto Agile "individui e interazioni *più che* processi e strumenti", i processi in un ambiente Agile saranno tipicamente documentati in modo leggero e non affronteranno tutti gli scenari possibili. La formazione delle persone, tuttavia, diventa un fattore critico di successo che richiede attenzione ed è quasi una contromisura per i processi leggeri. Un programma di formazione sulla qualità assicura che le persone coinvolte nel testing continuino a migliorare le proprie conoscenze e competenze in materia di testing e ad acquisire le necessarie conoscenze del dominio e altre conoscenze e competenze relative al testing. Poiché la qualità e il testing sono ormai responsabilità di un team, ci si aspetta che la formazione relativa al testing non venga fornita solo ai professionisti del testing, ma piuttosto a tutto il team Agile.

3.2.1 SG1 Stabilire una capacità organizzativa di addestramento al test

Il processo inizia con l'identificazione delle esigenze strategiche di formazione sui test dell'organizzazione. Oltre alle tradizionali esigenze di formazione per i tester, ad esempio sulle tecniche di progettazione dei test e sulle misure di copertura, è ora necessario acquisire conoscenze e competenze sulle specificità del testing Agile. Tra gli esempi vi sono il manifesto Agile e i suoi principi, i framework come Scrum, Kanban e XP, i concetti di test Agile (ad esempio, la piramide dei test e i quadranti di test), il Test Driven Development (TDD), il Behavior Driven Development (BDD), l'Acceptance Test-Driven Development (ATDD), lo sviluppo della User Story, compreso il supporto alla definizione dei criteri di accettazione e la revisione della loro testabilità, l'automazione dei test e i test esplorativi. Anche alcune aree di conoscenze e competenze tipiche della gestione dei test, come la pianificazione, la stima (ad esempio, il Planning Poker), il monitoraggio e l'analisi dei rischi (ad esempio, il Risk Poker), diventeranno aree che devono far parte del set di conoscenze e competenze della maggior parte dei tester che lavorano in un team Agile e non solo del test manager. L'identificazione delle esigenze strategiche di formazione sui test non si limita ai soli tester, ma è di estrema importanza specificare anche le esigenze di formazione sui test per gli altri membri dei team Agile.



Come in qualsiasi altra organizzazione, le esigenze di formazione sui test saranno tradotte in un piano di formazione (leggero), che eventualmente affronterà anche alcune esigenze specifiche di formazione sui test del progetto. Il piano di formazione deve essere rivisto e devono essere stabiliti degli impegni. Si noti che, soprattutto in Agile, alcune competenze vengono trasmesse in modo efficace ed efficiente attraverso veicoli informali (ad esempio, training-on-the-job, sessioni di formazione a pranzo, coaching e mentoring), mentre altre competenze richiedono una formazione formale. L'approccio appropriato per soddisfare le esigenze specifiche di formazione sui test deve essere determinato per ogni area di conoscenze e competenze, ad esempio, il training-on-the-job sui test esplorativi è un aspetto importante.

3.2.2 SG2 Fornire la formazione necessaria per i test

In linea di massima, le pratiche specifiche per l'SG2 sono le stesse in un'organizzazione tradizionale e Agile. Sulla base del piano di formazione dell'organizzazione, vengono identificati i membri del team che devono ricevere la formazione necessaria per poter svolgere efficacemente il proprio ruolo di test. La formazione viene quindi programmata, includendo le risorse necessarie, e condotta. Sembra un'affermazione ovvia, ma alcune

organizzazioni Agile faticano a trovare il tempo per la formazione. I team Agile sono sempre completamente impegnati e cercano di rispettare la scadenza di fine iterazione. Naturalmente, il tempo per la formazione deve essere considerato nella pianificazione dell'iterazione, in quanto il partecipante sarà meno disponibile per l'iterazione successiva.

Come già detto, in un'organizzazione Agile si utilizzano spesso strumenti di formazione informale. Ad esempio, oltre alla formazione formale, il mentoring e il coaching sul posto di lavoro sono tipicamente incoraggiati e utilizzati in modo continuativo. In effetti, il mentoring è una responsabilità di tutti ed è atteso a tutti i livelli dell'organizzazione. Il lavoro a coppie è un altro veicolo comunemente utilizzato dai team Agile per l'aggiornamento e il trasferimento delle conoscenze. Quando si utilizza Scrum, è responsabilità dello Scrum Master allenare il team alle pratiche di Scrum.

Verranno creati dei registri del test training condotto, sia informale che formale, e verrà valutata l'efficacia del test training. Una formazione (di test) a cui si è partecipato può essere valutata anche nell'ambito di una riunione retrospettiva in cui si discute se le conoscenze e le competenze dei partecipanti sono ora più adeguate per svolgere i Task di test.

3.3 Area di processo 3.3 Ciclo di vita del test e integrazione

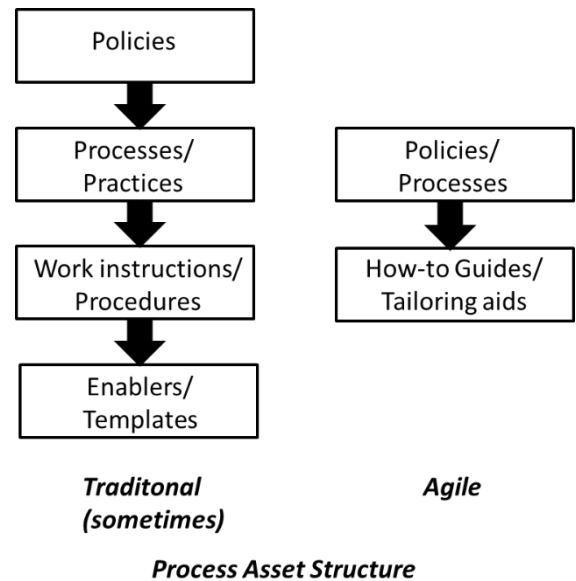
Lo scopo del Ciclo di vita del test e integrazione è quello di stabilire e mantenere un insieme utilizzabile di risorse organizzative per il processo di test (ad esempio, un ciclo di vita del test standard) e di standard per l'ambiente di lavoro e di integrare e sincronizzare il ciclo di vita del test con il ciclo di vita dello sviluppo. Il ciclo di vita integrato garantisce il coinvolgimento precoce del testing in un progetto. Lo scopo del ciclo di vita e dell'integrazione dei test è anche quello di definire un approccio coerente ai test su più livelli, basato sui rischi identificati e sulla strategia di test definita, e di fornire un Test Plan complessivo, basato sul ciclo di vita dei test definito.

3.3.1 SG 1 Stabilire gli asset del processo di test organizzativo

Un'importante responsabilità dell'organizzazione di test è quella di definire, documentare e mantenere un processo di test standard, in linea con la politica e gli obiettivi di test dell'organizzazione. Esso contiene, tra l'altro, una descrizione dei vari tipi e livelli di test da eseguire. Sebbene il TMMi non imponga processi di test dettagliati, definiti e documentati, spesso modelli mirati, elaborati con esempi, sono un ottimo approccio per stabilire un modo comune di lavorare. Si faccia riferimento anche ai tre pilastri della qualità e del testing Agile di Bob Galen, che identifica gli standard come Check List, modelli e archivi come uno dei pilastri del testing del software Agile [Galen]. Un modello identifica le attività necessarie per raccogliere informazioni e documentare i risultati impliciti nella sua struttura. Un processo non deve necessariamente essere una sequenza rigorosa di attività. Quando esistono delle dipendenze, queste possono essere catturate da note nel modello o da campi in un modulo che non sono accessibili finché non sono stati completati altri campi pre-requisiti. I modelli hanno un valore pratico e trasmettono l'intento reale di un processo. Inoltre, i modelli evitano le ambiguità che si trovano comunemente nei documenti di processo "verbosi". In un'organizzazione Agile, se disponete di processi efficaci, potreste aver bisogno di documentarli in modo leggero e forse di aggiungere qualche elemento. Alcune linee guida fondamentali spesso utilizzate nelle organizzazioni Agile quando si stabiliscono i processi:

- I "must" del processo sono confezionati separatamente dalle linee guida.
- Nessun processo supera le due pagine (obiettivo, regola soft)
- I processi non contengono informazioni sul "how-to" o sugli strumenti, a meno che non si sia deciso di imporli a tutti i progetti, indipendentemente dalle loro dimensioni e dalla loro portata.
- Le linee guida separate contengono opzioni di personalizzazione/pianificazione e informazioni "how-to". Si noti che con le informazioni "how-to" si può anche decidere di fare riferimento a un libro o a un documento esistente.

In molte grandi organizzazioni non è raro vedere quattro livelli di asset di processo: politiche, processi/pratiche, istruzioni/procedure di lavoro e strumenti/template. Questo non è dovuto a qualcosa che il TMMi richiede, ma al modo in cui molte grandi organizzazioni hanno scelto di implementare i loro asset di processo. Sebbene la scelta degli asset di processo dipenda da ciascuna organizzazione, la maggior parte delle organizzazioni Agile ha ritenuto che due livelli di asset di processo siano sufficienti. Questo si ottiene consolidando una dichiarazione di policy con la descrizione del processo associato, che racchiude "ciò che deve essere fatto" per portare a termine il processo. Il secondo livello contiene le linee guida su come eseguire il processo e adattarlo. Questo livello può essere visto come un aiuto per personalizzare il processo, dando autonomia al team per decidere come lavorare esattamente all'interno del quadro definito e di solito include modelli di supporto. Nella maggior parte delle organizzazioni Agile le procedure passo-passo sono sostituite da guide agli strumenti e da formazione/mentoring. Come già detto, un modello, come un modello di Test Chart, un modello di scheda di sessione di test o un modello di Test Plan, può fungere da processo con le attività di processo richieste implicite nel modello. Questa è una tecnica comunemente osservata per sviluppare descrizioni efficaci dei processi Agile. [McMahon]



L'insieme dei processi di test standard dell'organizzazione può essere adattato dai progetti per creare i loro processi specifici definiti. Nel contesto dell'SP 1.3 Stabilire criteri e linee guida per l'adattamento, la maggior parte delle organizzazioni oggi utilizza il cosiddetto approccio tailor down. Questo approccio richiede che le persone si impegnino a spiegare perché qualcosa non è necessario. Questo aggiunge la maggior quantità di lavoro per i progetti più semplici, il che non è un approccio favorevole ad Agile. Quando si fa il tailor down, fino a che punto ci si può spingere? Esiste un minimo definito? Tuttavia, se si parte da un set minimo in cui il set minimo è ciò che tutti devono fare, si elimina il rischio di personalizzare un "must do". Un approccio di tipo tailoring up è molto più coerente con gli approcci Agile e completamente conforme a TMMi. Il tailoring è un meccanismo estremamente potente per le organizzazioni Agile che desiderano mantenere la propria agilità o per le organizzazioni ad alta maturità (test) che cercano di aumentare la propria agilità. Un insieme di semplici criteri potrebbe aiutare a porsi le domande giuste quando si fa il tailoring. Senza criteri di aiuto, alcuni tendono a fare troppo tailoring.

Per supportare l'implementazione del processo di test standard, verrà creata e mantenuta una libreria di asset del processo di test, spesso su un wiki, in cui i tester possono trovare, ad esempio, modelli, best practice e Check List, che li aiuteranno nel loro lavoro quotidiano. La creazione di una libreria di asset per il processo di test ha senso anche in un contesto Agile, purché i team ottengano un valore aggiunto nell'utilizzo degli asset forniti durante le loro attività di test. Più o meno lo stesso ragionamento vale per un database del processo di test in cui vengono raccolti e messi a disposizione dei team i dati di test, ad esempio stime, copertura, misure di qualità. Assicuratevi che tutto ciò che viene raccolto e reso disponibile per consentire l'apprendimento e la condivisione delle esperienze tra i team abbia un valore aggiunto per i team.

Naturalmente, anche in un contesto Agile gli standard dell'ambiente di lavoro vengono utilizzati per guidare la creazione degli ambienti di lavoro del progetto.

3.3.2 SG 2 Integrare i modelli del ciclo di vita del test con i modelli di sviluppo

Il modello standard del ciclo di vita del test definisce le attività e i risultati principali per i vari livelli di test. Il modello standard del ciclo di vita del test è allineato con i modelli del ciclo di vita dello sviluppo per integrare le attività di test in termini di fasi, milestone, deliverable e attività. L'integrazione del ciclo di vita avviene in modo tale da garantire un coinvolgimento precoce del testing nei progetti. Tutto questo suona come se non ci fosse nulla da fare nel caso in cui sia stato adottato un ciclo di vita Agile; tutto è già stato preso in considerazione. In Agile, lo sviluppo e il testing dovrebbero essere totalmente integrati nel ciclo di vita e il testing dovrebbe avvenire il più presto possibile.

Per quanto questo sia vero in molte organizzazioni che utilizzano Agile, ci sono anche organizzazioni che lottano con questo aspetto. La loro iterazione assomiglia a un mini-ciclo di vita a V per lo sviluppo, dove i test iniziano solo verso la fine dell'iterazione, sono spesso eseguiti sotto pressione e a volte persino tagliati. Ovviamente non si tratta di un'implementazione di un ciclo di vita integrato di sviluppo e test Agile, come dovrebbe essere. Tuttavia, dimostra che per alcune organizzazioni "Agile-like" questo obiettivo specifico è un'area di miglioramento. È importante creare un'intesa comune definendo il modo in cui il testing può essere coinvolto ed eseguito all'inizio dell'iterazione.

Questo obiettivo specifico è quindi applicabile anche a un'organizzazione che si occupa di sviluppo software Agile. Se fatto in modo appropriato, lo sviluppo e il testing dovrebbero essere completamente integrati e c'è un'intesa comune su come farlo. In questi casi, non è necessario fare nulla di più. In altre circostanze, come quelle descritte in precedenza, questo obiettivo specifico è un'area di miglioramento molto importante e dovrebbe portare a un migliore allineamento delle attività di sviluppo e di test eseguite in un'iterazione.

3.3.3 SG 3 Stabilire un Test Plan principale

Al livello 3 del TMMi, il collaudo si occupa della pianificazione del collaudo principale, che riguarda il coordinamento dei Task, delle responsabilità e dell'approccio al collaudo tra tutti i livelli di collaudo. Ciò evita inutili ridondanze o omissioni di test tra i vari livelli di test e può aumentare significativamente l'efficienza e la qualità del processo di test complessivo. Il master test plan descrive l'applicazione della strategia di test per un particolare progetto, compresi i livelli specifici da eseguire e la relazione tra tali livelli. A livello TMMi 2, la pianificazione (dei test) è già affrontata sia a livello di rilascio che di iterazione come parte dell'area di processo Pianificazione dei test. Un team Agile tipicamente esegue più livelli di test, che dovrebbero essere tutti affrontati in modo appropriato come parte della pianificazione del rilascio e dell'iterazione.

Tuttavia, alcuni progetti che lavorano secondo un modello di ciclo di vita Agile hanno adottato un approccio in qualche modo ibrido, nel senso che oltre ai test all'interno di un team Agile, alcuni test sono organizzati al di fuori del team. In tali circostanze, un master test plan definirà e gestirà il rapporto tra i team Agile e i livelli di test eseguiti al di fuori del loro ambito, ad esempio test non funzionali specifici (sicurezza, prestazioni), test di integrazione hardware/software, test di integrazione del sistema o beta test. A questo obiettivo specifico saranno tipicamente applicabili anche pratiche specifiche correlate. Soprattutto nei progetti più grandi e con la crescita delle organizzazioni, la documentazione di un piano generale di test e delle relative decisioni in modo coerente inizia ad avere un valore maggiore. Questo è particolarmente vero quando un'organizzazione utilizza framework Agile scalati.

Nel caso in cui tutti i test si svolgano con i team Agile, un Test Plan specifico non ha alcun valore aggiunto e quindi questo obiettivo di TMMi è probabilmente non rilevante in questi casi. In questo caso, la pianificazione a livello di rilascio e di iterazione dovrebbe coprire tutti gli aspetti di test necessari.

3.4 Area di processo 3.4 Test non funzionali

Lo scopo dell'area di processo Test non funzionali è quello di migliorare le capacità del processo di test per includere i test non funzionali durante la pianificazione, la progettazione e l'esecuzione dei test.

L'importanza dei test non funzionali è indipendente dal ciclo di vita applicato. Il prodotto che si sta sviluppando determinerà quali aspetti non funzionali sono rilevanti per il test. Questo vale ovviamente sia per i modelli di ciclo di vita sequenziali che per i modelli di sviluppo Agile. Il test non funzionale è quindi applicabile anche allo sviluppo software Agile.



Tuttavia, a seconda della natura degli aspetti non funzionali e dell'approccio di test utilizzato, alcuni test non funzionali, ad esempio le performance e l'affidabilità, se testati in modo tradizionale, non possono essere testati in una breve iterazione. Poiché questi test non funzionali, a seconda dei metodi e delle tecniche utilizzate, possono richiedere settimane per essere impostati, organizzati ed eseguiti, il loro tempo di esecuzione è meno applicabile al cuore delle iterazioni. È quindi spesso necessario un approccio diverso al test degli attributi non funzionali. Questo è uno dei cambiamenti fondamentali in meglio con Agile, in quanto gli attributi di qualità vengono testati già nelle prime fasi

dell'iterazione, e non lasciati alla fine come in un modello di ciclo di vita tradizionale. Non tutti le componenti/funzioni possono essere disponibili fin dalla prima iterazione per eseguire un test non funzionale completo, ma i risultati di questo test possono essere utilizzati per fornire indicazioni precoci sui problemi di qualità.

Le storie degli utenti devono riguardare sia gli elementi funzionali che quelli non funzionali, per garantire che venga sviluppato il prodotto giusto per gli utenti e i clienti. Le caratteristiche di qualità ISO 25010 possono aiutare a strutturare i requisiti e i tester dovrebbero considerare questi elementi non funzionali [ISO25010].

3.4.1 SG1 Esecuzione di una valutazione del rischio non funzionale del prodotto

Le sessioni di rischio del prodotto per i progetti Agile, identificate e descritte nell'obiettivo specifico SG1 Eseguire la valutazione dei rischi come parte dell'area di processo Pianificazione dei test, saranno ora esplicitamente estese per includere anche gli aspetti e i rischi non funzionali. A livello di release la valutazione dei rischi, ora anche per i test non funzionali, può essere eseguita sulla base della visione del prodotto. A livello di iterazione viene eseguita utilizzando come input principale le User Story che definiscono i requisiti non funzionali.

Preferibilmente tutti i membri del team, compreso il Product Owner, ed eventualmente altri stakeholder dovrebbero partecipare alle sessioni di rischio del prodotto. Per alcune aree non funzionali può essere necessaria l'assistenza di specialisti. Le sessioni sui rischi del prodotto si concluderanno in genere con un elenco documentato di rischi del prodotto (non funzionali) prioritari. Come indicato nell'area di processo Pianificazione dei test, nei progetti agili sia il processo utilizzato che la documentazione risultante saranno molto più leggeri rispetto a un progetto tradizionale che segue un modello di ciclo di vita sequenziale.

3.4.2 SG 2 Stabilire un approccio di test non funzionale

Viene definito un approccio di test per le caratteristiche qualitative non funzionali rilevanti, al fine di mitigare i rischi non funzionali del prodotto identificati e prioritari. Per una specifica iterazione, le caratteristiche non funzionali da testare vengono identificate durante la pianificazione dell'iterazione. L'elenco prioritario delle caratteristiche non funzionali da testare si riferisce tipicamente alle User Story da testare in questa iterazione. Durante l'iterazione possono emergere nuovi rischi non funzionali del prodotto che richiedono ulteriori test. Le questioni come i nuovi rischi non funzionali del prodotto che richiedono test aggiuntivi vengono tipicamente discusse durante le riunioni giornaliere di standup.

L'approccio al test non funzionale definito a livello di iterazione per mitigare i rischi non funzionali riguarderà in genere l'identificazione di metodi e tecniche di test non funzionali appropriati in base al livello e al tipo di rischi non funzionali. Di solito si occupa anche dell'uso di strumenti di supporto e dell'approccio all'automazione dei test per i test non funzionali. L'approccio per il test non funzionale a livello di rilascio sarà a un livello molto più alto e dovrebbe basarsi sulla strategia di test definita a livello di programma o di organizzazione e anche sul Test Plan principale definito, se esiste. (Fare riferimento a SG3 Stabilire un Test Plan principale all'interno dell'area di processo Ciclo di vita del test e integrazione per le motivazioni che spingono a stabilire o meno un Test Plan principale). Sia l'approccio di test non funzionale a livello di rilascio che di iterazione fanno parte dell'approccio di test complessivo e saranno conservati o visualizzati sul wiki del team/progetto.

I criteri di uscita non funzionali fanno parte della cosiddetta Definition of Done (DoD). È importante che la DoD contenga criteri specifici relativi ai test non funzionali, ad esempio il tempo medio tra i guasti (MTBF) o "le pagine

web front-end sono state testate per l'elenco dei 10 rischi principali di OWASP". L'iterazione deve portare all'implementazione dell'insieme concordato di User Story non funzionali (compresi i relativi criteri di accettazione) e soddisfare i criteri di uscita (test) non funzionali definiti nel DoD. Si noti che gli attributi non funzionali possono anche far parte dei criteri di accettazione delle User Story funzionali e non devono necessariamente essere specificati come User Story non funzionali separate. Non solo esiste una Definition of Done a livello di iterazione, ma spesso esiste anche una DoD a livello di release che abbraccia più iterazioni. Anche la DoD a livello di release può avere criteri non funzionali correlati.

3.4.3 SG 3 Esecuzione di analisi e progettazione di test non funzionali

Questo obiettivo specifico segue in gran parte le stesse pratiche, ma ora da una prospettiva non funzionale, come per l'obiettivo specifico SG1 Eseguire l'analisi e la progettazione dei test utilizzando le tecniche di progettazione dei test dell'area di processo Progettazione ed esecuzione dei test. Durante l'analisi e la progettazione dei test, l'approccio ai test non funzionali viene tradotto in condizioni di test e test tangibili. In Agile, l'analisi e la progettazione dei test e l'esecuzione dei test sono attività che si supportano a vicenda e che di solito si svolgono in parallelo nel corso di un'iterazione. Questo vale anche per la maggior parte dei test non funzionali. L'analisi dei test non funzionali non è quindi un'attività esplicitamente separata, ma piuttosto un'attività implicita che i tester svolgono come parte del loro ruolo all'interno dello sviluppo collaborativo delle User Story.

Sulla base dell'analisi delle User Story non funzionali, vengono identificate le condizioni di test non funzionali. Le condizioni di test sono fondamentalmente un'identificazione di "cose" che devono essere testate/coperte. Se i criteri di accettazione definiti sono sufficientemente dettagliati e chiari, in genere assumono il ruolo delle condizioni di test tradizionali. I criteri di accettazione vengono successivamente tradotti in test non funzionali. Con i test non funzionali è spesso vantaggioso eseguire l'analisi dei test a un livello superiore rispetto alle sole User Story. Ad esempio, analizzando una caratteristica o un'Epic o un insieme di storie per identificare le condizioni di test non funzionali che sono più astratte di quelle a livello di User Story e che coprono anche più User Story. Con l'applicazione del principio test-first in Agile, i test non funzionali che coprono l'insieme delle condizioni di test non funzionali saranno identificati (e possibilmente automatizzati) prima, o almeno in parallelo, allo sviluppo del codice.

Per la maggior parte dei test manuali non funzionali, i test saranno identificati/raffinati man mano che il team procede con l'esecuzione dei test non funzionali. Il più delle volte i test non sono documentati in modo così dettagliato come nei progetti tradizionali, ma piuttosto in un formato di idee di test quando si utilizzano test esplorativi. Per le aree non funzionali complesse e critiche, un approccio più tradizionale alla progettazione dei test e allo sviluppo dei casi di test utilizzando tecniche formali di progettazione dei test non funzionali può essere il modo migliore per coprire i rischi. Tuttavia, anche con questo approccio la quantità di documentazione sarà limitata rispetto ai test non funzionali in un ambiente di ciclo di vita sequenziale tradizionale. La priorità dei test non funzionali segue tipicamente la priorità della User Story che stanno coprendo. Tuttavia, la priorità può essere determinata anche dal tempo necessario per preparare ed eseguire un determinato test non funzionale.

Vengono identificati i dati di test specifici necessari per supportare l'esecuzione dei test non funzionali. In Agile i dati di test necessari in genere non vengono prima specificati completamente in un documento di specifiche di test. Quando vengono specificati, spesso vengono registrati come parte della User Story. Tuttavia, a condizione che siano disponibili gli strumenti e/o le funzionalità necessarie, i dati di test necessari per supportare i test non funzionali sono spesso creati istantaneamente per consentire un rapido avvio dell'esecuzione dei test manuali non funzionali. Al contrario, per i test non funzionali automatizzati, i dati devono essere specificati in anticipo.

È necessario stabilire e mantenere la tracciabilità tra i requisiti non funzionali, le condizioni di test e i test. I team devono assicurarsi di aver coperto le varie User Story non funzionali e i criteri di accettazione come parte dei loro test. In molte organizzazioni Agile, le User Story sono la base per lo sviluppo di una serie di criteri di accettazione correlati e, successivamente, dei test. Con questo approccio è possibile ottenere una tracciabilità orizzontale dai requisiti ai test.

3.4.4 SG 4 Eseguire l'implementazione di test non funzionali

L'implementazione dei test consiste nel predisporre tutto ciò che è necessario per avviare l'esecuzione dei test. In genere lo sviluppo della documentazione di test, ad esempio le procedure di test, per supportare l'esecuzione dei test è ridotto al minimo. Piuttosto, vengono sviluppati script di test automatizzati (di regressione) e viene assegnata loro una priorità.

L'implementazione di test non funzionali seguirà molte delle pratiche già descritte nell'obiettivo specifico SG2 Esecuzione di test come parte dell'area di processo Progettazione ed esecuzione dei test.

Ciò che deve essere fatto in modo specifico e il modo in cui viene eseguita l'implementazione dei test non funzionali dipende in larga misura dall'approccio definito, dalle tecniche utilizzate e da quali caratteristiche non funzionali devono essere testate ed a quale livello. Ad esempio, i test esplorativi, che richiedono una minore preparazione, possono essere adatti per i test di usabilità, ma spesso sono meno adatti per i test approfonditi di affidabilità e prestazioni.

Per alcune caratteristiche qualitative non funzionali, la disponibilità di dati di test è essenziale e deve essere creata durante l'implementazione del test. Questo è in gran parte lo stesso dei progetti tradizionali, a parte il fatto che forse sono documentati in modo sintetico, consentendo comunque il riutilizzo per i test di regressione.

Naturalmente, l'implementazione e la preparazione del test inizieranno il prima possibile, parallelamente alle altre attività (di test). Non si tratta di una fase separata, ma piuttosto di un insieme di attività (elencate nella task board) che devono essere eseguite per consentire un'esecuzione efficiente ed efficace dei test.

3.4.5 SG 5 Esecuzione del test non funzionale

Come per il precedente obiettivo di quest'area di processo, ci riferiremo in gran parte al relativo obiettivo specifico SG 3 Esecuzione dei test, parte dell'area di processo Progettazione ed esecuzione dei test discussa in precedenza in questo documento. Le pratiche per l'esecuzione di test non funzionali, la segnalazione di incidenti di test e la scrittura di Test Log in un ambiente Agile sono fondamentalmente le stesse dei test funzionali in un ambiente Agile. In genere, la documentazione è molto meno intensa che in un progetto tradizionale. Spesso non vengono prodotte procedure e Test Log dettagliati.

L'esecuzione dei test non funzionali avviene in linea con le priorità definite durante la pianificazione dell'iterazione. Alcuni test possono essere eseguiti utilizzando una procedura di test documentata, ma in genere anche molti test non funzionali vengono eseguiti utilizzando test esplorativi e basati su sessioni come struttura. Con lo sviluppo iterativo aumenta la necessità di organizzare e strutturare i test di regressione. A volte questo viene fatto manualmente, ma di preferenza si ricorre a test di regressione automatizzati e a strumenti di supporto. I test di regressione, ovviamente, si applicano anche agli aspetti non funzionali del sistema che sono stati identificati come importanti da testare.

Gli incidenti non funzionali riscontrati durante i test possono essere registrati e segnalati dal team. In genere nei progetti Agile si discute se tutti gli incidenti riscontrati debbano essere registrati. Alcuni team registrano solo gli incidenti che sfuggono alle iterazioni, altri li registrano se non possono essere risolti oggi, altri ancora registrano solo gli incidenti ad alta priorità. Se non tutti gli incidenti trovati vengono registrati, devono essere disponibili dei criteri per determinare quali incidenti devono essere registrati e quali no. A volte gli incidenti vengono registrati sulla task board di Agile, come un adesivo su un'attività o come un'attività separata, visualizzando che stanno bloccando il completamento di una storia e delle sue attività. Alcuni scelgono di registrare e documentare gli incidenti riscontrati utilizzando uno strumento, ad esempio uno strumento di gestione dei difetti o di tracciamento dei difetti. Tale strumento deve essere utilizzato nel modo più snello possibile e non deve costringere a presentare elementi dei rapporti sugli incidenti di test non funzionali che non hanno alcun valore aggiunto o ne hanno troppo poco.

È considerata una buona pratica anche nei team Agile registrare i dati durante l'esecuzione dei test non funzionali per determinare se gli elementi testati soddisfano i criteri di accettazione definiti e possono essere effettivamente etichettati come "fatti". Le informazioni registrate dovrebbero essere loggate e/o riassunte in qualche forma di strumento di gestione dello stato (ad esempio, strumenti di gestione dei test, strumenti di gestione delle attività, task board), in modo da rendere facile per il team e gli stakeholder comprendere lo stato attuale di tutti i test eseguiti.

3.5 Area di processo 3.2 Peer Review

Lo scopo del processo di Peer Review è quello di verificare che i prodotti di lavoro soddisfino i requisiti specificati e di eliminare tempestivamente ed efficacemente i difetti dai prodotti di lavoro selezionati. Un importante corollario è lo sviluppo di una migliore comprensione dei prodotti di lavoro e dei difetti che potrebbero essere evitati.

3.5.1 SG 1 Stabilire un approccio di Peer Review

I team Agile di solito non effettuano Peer Review formali, nel senso che di solito non hanno un singolo momento definito in cui le persone si incontrano per fornire un feedback su un prodotto. Tuttavia, raggiungono l'obiettivo delle Peer Review effettuando continue Peer Review meno formali nel corso dello sviluppo. Questa è una pratica comune a molte organizzazioni Agile. Tuttavia, è necessaria una disciplina nella conduzione di queste attività. Questo obiettivo TMMi riguarda le pratiche per stabilire un approccio di revisione tra pari all'interno di un progetto. Un approccio alla revisione definisce come, dove e quando devono svolgersi le attività di revisione e se queste attività sono formali o informali. Stabilire un approccio di peer review è applicabile anche ai progetti Agile, ma in genere le tecniche di revisione applicate e il modo in cui le revisioni sono organizzate sono molto diversi.

Esempi di peer review tipicamente eseguiti nell'ambito di progetti Agile:

- avere sessioni di perfezionamento/elaborazione delle specifiche (ad esempio, User Story) con il team e gli stakeholder aziendali su base regolare nel corso di un'iterazione;
- riunioni quotidiane con gli altri membri del team per discutere apertamente dei prodotti di lavoro, ad esempio codice o test, in fase di sviluppo e fornire feedback.
- la dimostrazione dei prodotti in anticipo e spesso ai clienti, almeno alla fine di un'iterazione durante la revisione dell'iterazione;

Le specifiche scadenti sono spesso una delle ragioni principali del fallimento di un progetto. I problemi di specifiche possono derivare dalla mancanza di comprensione delle reali esigenze degli utenti, dall'assenza di una visione globale del sistema, da caratteristiche ridondanti o contraddittorie e da altri errori di comunicazione. Nello sviluppo Agile, le storie degli utenti vengono scritte per catturare i requisiti dal punto di vista dei rappresentanti del business, degli sviluppatori e dei tester. Nello sviluppo sequenziale, la visione condivisa di una funzionalità viene realizzata attraverso revisioni formali dopo la stesura dei requisiti; nello sviluppo Agile, questa visione condivisa viene realizzata attraverso frequenti revisioni informali durante la definizione dei requisiti. Queste sessioni di revisione informale sono spesso definite sessioni di affinamento del backlog o di backlog grooming. Durante le riunioni di affinamento, il rappresentante del business e il team di sviluppo (e gli stakeholder, se disponibili) utilizzano tecniche di revisione per trovare il livello di dettaglio necessario per l'implementazione e per chiarire le questioni aperte.

La revisione quasi continua da parte del team dei prodotti di lavoro in fase di sviluppo fa parte dell'approccio Whole Team. Vengono eseguite con l'intento di identificare precocemente i difetti e di individuare le opportunità di miglioramento. Metodi e tecniche agili come XP o pairing includono anche le Peer Review come pratica fondamentale per creare cicli di feedback per i team. Naturalmente, in caso di elevata complessità o rischio, il team può scegliere di applicare una tecnica di revisione semi-formale o formale, ad esempio l'ispezione. In questi casi c'è una chiara ragione per dedicare sforzi a una revisione più formale e applicare un metodo di lavoro più disciplinato. Una parte dei criteri di approccio alla revisione può essere definita quando utilizzare una tecnica di revisione più formale.

Mentre l'ingegneria dei requisiti enfatizza la convalida dei requisiti attraverso metodi come revisioni informali, walkthrough, ispezioni o letture prospettive, i metodi Agile si sforzano piuttosto di convalidare i requisiti attraverso feedback frequenti e precoci per implementare rapidamente incrementi di prodotto di valore. La necessità di una validazione formale precoce è ridotta dalla presentazione di risultati rapidi sotto forma di incrementi di prodotto integrati. Se l'incremento non soddisfa completamente i requisiti degli stakeholder, il delta viene reinserito nel backlog del prodotto sotto forma di nuovi requisiti e prioritizzato con tutti gli altri elementi del backlog. Una dimostrazione di ciò che è stato effettivamente costruito durante un'iterazione è semplicemente un modo molto efficace per stimolare una conversazione basata sulla validazione di qualcosa di concreto. Niente mette a fuoco la conversazione come la possibilità di vedere effettivamente come funziona qualcosa. La dimostrazione è un'attività che si svolge durante la revisione dell'iterazione. Le caratteristiche vengono mostrate e discusse con gli stakeholder, e vengono

apportati i necessari adattamenti al backlog del prodotto o al piano di rilascio per riflettere le nuove conoscenze acquisite durante la discussione.

3.5.2 SG 2 Eseguire Peer Review

Naturalmente, come ogni approccio, non deve esistere solo come approccio definito e concordato, ma deve essere rispettato. Il team deve dedicare una quantità sostanziale di tempo alle sessioni di affinamento del backlog durante un'iterazione, applicare revisioni informali (e possibilmente formali) come parte della routine quotidiana e fare dimostrazioni agli stakeholder su base regolare. Le dimostrazioni agli stakeholder/clienti sono previste almeno alla fine di ogni iterazione.

I criteri di ingresso sono tipicamente applicabili solo alle revisioni formali e quindi probabilmente sono meno o non rilevanti nei progetti Agile. Tuttavia, l'uso dei criteri di uscita è applicabile e ha un valore aggiunto. INVEST [Wake] è un esempio di un insieme di criteri di uscita a cui si fa comunemente riferimento nei progetti Agile durante la revisione e l'aggiornamento delle User Story. INVEST è un acronimo che racchiude i seguenti concetti che costituiscono una buona User Story:

- Indipendente
- Trattabile
- Valutabile
- Stimabile
- Piccolo
- Testabile.



Naturalmente esistono altri criteri di uscita legati alla qualità che possono essere utilizzati in modo vantaggioso.

È importante che il tester, facendo parte del team Agile, partecipi alle sessioni di revisione. Questo è particolarmente vero per le sessioni in cui vengono discussi e rivisti i prodotti di lavoro che vengono utilizzati come base per i test nel corso dell'iterazione, ad esempio nelle sessioni di rifinitura del backlog. Si raccomanda la presenza di almeno uno sviluppatore e di un tester durante la rifinitura del backlog, per garantire la presenza di punti di vista alternativi sul sistema. In genere, la prospettiva unica del tester migliorerà la User Story identificando i dettagli mancanti o i requisiti non funzionali. In particolare, il tester può contribuire all'identificazione e alla definizione dei criteri di accettazione per una determinata User Story. Il tester contribuisce ponendo ai rappresentanti aziendali domande aperte del tipo "e se?" sulla User Story, proponendo modi per testare la User Story e confermando i criteri di accettazione.

La pratica specifica 2.3 Analizzare i dati delle Peer Review è particolarmente importante per le revisioni formali. Con le revisioni formali si spende molto impegno, per garantire che l'impegno sia speso in modo efficiente ed efficace si raccolgono i dati delle peer review e si comunicano al team per imparare e mettere a punto il processo di revisione. Come detto, le revisioni formali, ad esempio le ispezioni, sono meno comuni nei progetti Agile. Mentre la raccolta di dati è una parte essenziale delle revisioni formali, è molto meno comune nelle revisioni informali. Di conseguenza, si può sostenere che la raccolta, l'analisi e la comunicazione dettagliata dei dati della peer review siano meno o addirittura non rilevanti in un contesto Agile. Nel contesto della decisione di raccogliere dati durante le peer review, ponetevi le seguenti domande:

- Chi utilizzerà questi dati se vengono raccolti?
- Che relazione hanno questi dati con i nostri obiettivi aziendali?

Se nessuno utilizzerà i dati in modo significativo, allora non sprecate risorse preziose per raccoglierci. In conclusione, per la maggior parte dei progetti Agile la pratica specifica 2.3 Analizzare i dati delle Peer Review sarà considerata probabilmente non rilevante. Si noti che alcuni dati di base sulla revisione saranno comunque raccolti e utilizzati nel contesto delle pratiche generiche, ad esempio la GP 2.8 Monitoraggio e controllo del processo.

4 TMMi Livello 4 Misurato

Come affermato in precedenza in questo documento, è importante ricordare che esiste una interruzione naturale tra il livello 3 del TMMi e i livelli 4 e 5 del TMMi. Ci sono ancora molte controversie sul valore del passaggio di un'organizzazione ai livelli 4 e 5 di TMMi. In particolare, alle organizzazioni Agile viene raccomandato di scegliere criticamente e di adottare solo le pratiche dei livelli TMMi 4 e 5 che sono importanti e hanno un valore aggiunto. Sebbene il TMMi sia completo, per avere successo le organizzazioni devono identificare le pratiche di testing e i miglioramenti chiave che richiedono attenzione. È interessante notare che Jeff Sutherland, cofondatore di Scrum, ha pubblicato un documento sul valore dell'utilizzo di Scrum insieme a pratiche di livello di maturità superiore [Sutherland et al]. Di seguito verranno fornite informazioni sui livelli 4 e 5 di TMMi e su come sia possibile ottenere il valore di questi livelli più elevati di TMMi e delle relative pratiche utilizzandoli in modo meno formale insieme a un approccio Agile.

In primo luogo, in contrasto con quanto affermato, si può anche considerare di utilizzare selettivamente le pratiche TMMi di livello 4 e 5 con le tecniche Agile per raggiungere gli obiettivi aziendali chiave, mentre si cerca di ottenere il livello TMMi 2 o 3. Non è necessario aspettare e non si deve farlo. Siate flessibili e non rimanete bloccati sulle pratiche di test TMMi di livello 2 e 3. Utilizzate il modello in base ai vostri driver aziendali e utilizzate le aree di processo, gli obiettivi e le pratiche che hanno più valore. Pertanto, utilizzate il TMMi in una modalità più continua e non utilizzate i livelli di maturità in modo troppo rigido.

4.1 Area di processo 4.1 Misurazione dei test

Lo scopo della misurazione dei test è quello di identificare, raccogliere, analizzare e applicare misure per supportare un'organizzazione nella valutazione oggettiva dell'efficacia e dell'efficienza del processo di test, della produttività del team addetto ai test, della qualità del prodotto risultante e dei risultati del miglioramento dei test. In questo modo, l'organizzazione di test svilupperà e sosterrà una capacità di misurazione dei test utilizzata per supportare le esigenze informative della direzione.

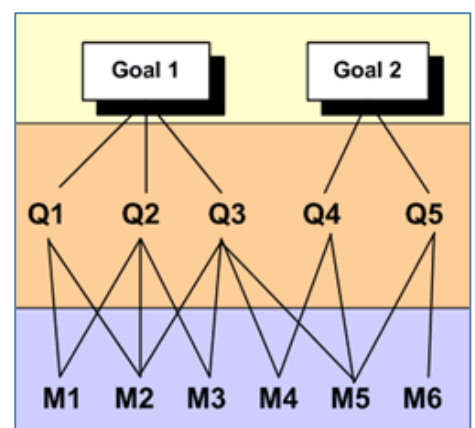
Gli approcci Agile aiutano a ottenere l'impegno di coloro che devono eseguire e misurare accuratamente il lavoro nei progetti. Tuttavia, gli approcci Agile non tendono a fornire molto supporto quando si tratta di fare un passo indietro per garantire che le misure giuste siano utilizzate in tutta l'organizzazione e che tali misure siano allineate con gli obiettivi di misurazione derivati dalle esigenze di business dell'organizzazione. Questi sono i punti di forza che l'area del processo di misurazione dei test TMMi può apportare alle organizzazioni Agile.

4.1.1 SG 1 Allineare le attività di misurazione e analisi dei test

Troppo spesso i programmi di misurazione sembrano essere finalizzati a raccogliere il maggior numero di dati possibile. Un programma di misurazione deve essere molto mirato e raccogliere solo gli elementi importanti. Iniziate definendo obiettivi chiari per il programma di misurazione e rivedeteli regolarmente, ad esempio: "Dovrebbe permetterci di avere più successo nei test dei nostri progetti". Successivamente, discutete con gli stakeholder quali metriche di test supporteranno realmente questi obiettivi. Sulla base di questo brainstorming verrà definito un insieme limitato di metriche fondamentali.

Il Goal-Question-Metric (GQM) [Van Solingen e Berghout] è un metodo molto pratico che supporta questo approccio. Sono necessarie misure specifiche e pertinenti al contesto, basate sulle esigenze aziendali. Le metriche standard aziendali sono spesso insufficienti per un reale miglioramento dei processi (test). Lo scopo delle misurazioni è guidare le decisioni. Utilizzate piccoli team responsabilizzati per ricavare misure significative e poi rivedetele e perfezionatele in cicli brevi.

Con un programma mirato di misurazione dei test che parte dal valore aggiunto (aziendale), l'area di processo TMMi Test Measurement è totalmente in linea con il principio Agile della semplicità. Assicurarsi che all'interno dell'organizzazione esista una strategia snella di misurazione dei test e di raccolta dei dati. Vale la pena sottolineare



che è possibile far evolvere un programma di misurazione dei test iniziando con pochi obiettivi di misurazione mirati e poche misure chiave, per poi ampliarli o modificarli in seguito, in base all'evoluzione e al cambiamento delle esigenze aziendali.

Si noti che nell'ambito di Agile l'attenzione si concentra maggiormente sul team e sul pensiero sistemico. Ciò può comportare un corrispondente allargamento di alcune metriche al team e al sistema complessivo, anziché limitarsi alle sole specificità del test. Questo può portare a maggiori sfide nella fase di analisi delle misurazioni.

Naturalmente le metriche che verranno definite e per le quali verranno raccolti i dati dipenderanno dagli obiettivi di misurazione, ma alcuni esempi di metriche di test tipiche per i team agili sono:

- Tempo di ciclo dei difetti; i team Agile dovrebbero sforzarsi di risolvere i difetti il più rapidamente possibile. Infatti, uno degli obiettivi principali dell'approccio collaborativo Agile è quello di risolvere i difetti più rapidamente, in modo da rilasciare il software prima possibile.
- Defect spill-over (numero di difetti rimandati a una release futura); i team Agile mirano a produrre software funzionante a ogni iterazione. Il Defect spillover misura i difetti che non vengono risolti durante una determinata iterazione o sprint, semplicemente contando i difetti rimasti alla fine di ogni sprint o iterazione. In alcune organizzazioni Agile questo viene definito debito tecnico.
- Numero di difetti riscontrati durante un'iterazione
- Numero di difetti riscontrati dopo il rilascio alla produzione/clienti
- Numero di difetti trovati da persone esterne al team
- Numero di richieste di assistenza clienti
- Percentuale di copertura dei test automatizzati
- Percentuale di copertura del codice.

4.1.2 SG 2 Fornire il risultato della misurazione del test

Una volta definite le misure rilevanti in base alle esigenze aziendali, le pratiche specifiche nell'ambito di questo obiettivo sono in genere tutte applicabili in modo sostanzialmente analogo a quelle di un'organizzazione tradizionale. L'unica differenza è che il tutto sarà fatto con una mentalità diversa, mantenendo le pratiche il più possibile snelle.

I dati di misura necessari per i test vengono raccolti e controllati per verificarne la completezza e l'integrità; successivamente i dati vengono analizzati come previsto e i risultati vengono comunicati a tutte le parti interessate.

4.2 Area di processo 4.2 Valutazione della qualità del prodotto

Lo scopo della valutazione della qualità dei prodotti è quello di sviluppare una comprensione quantitativa della qualità dei prodotti e quindi di sostenere il raggiungimento degli obiettivi di qualità di progetti specifici.

4.2.1 SG 1 Vengono stabiliti obiettivi di progetto misurabili per la qualità del prodotto e le relative priorità.

In sostanza, non c'è alcuna differenza nell'identificare e stabilire obiettivi prioritari e misurabili per la qualità del prodotto in un contesto Agile rispetto a quanto avviene in un ambiente tradizionale. L'obiettivo generale è quello di contribuire a soddisfare le esigenze e i desideri dei clienti e degli utenti finali di prodotti di qualità. Esistono diverse opzioni per raggiungere questo obiettivo in un contesto Agile. Alcuni usano la Definition-of-Done per definire obiettivi misurabili per la qualità del prodotto, altri usano caratteristiche, Epic o User Story per definirli, utilizzando anche criteri di accettazione per renderli misurabili. La scelta (il modo di lavorare) dipende, tra l'altro, dalla natura dell'obiettivo di qualità del prodotto. Per una panoramica degli attributi di qualità si rimanda a [ISO25010].

La Definition of Done (DoD) è una Check List completa delle attività e/o dei prodotti di lavoro necessari al team per garantire che vengano consegnate solo User Story veramente fatte, non solo in termini di funzionalità ma anche di qualità del prodotto. Il vantaggio di dichiarare gli obiettivi di qualità del prodotto all'interno del DoD è che li rende visibili e quindi perseguibili da tutto il team man mano che gli incrementi vengono sviluppati. È ovvio che gli obiettivi di qualità devono essere globalmente applicabili all'applicazione per poter essere menzionati nel DoD. Se gli

obiettivi di qualità non sono applicabili a livello globale, probabilmente il modo migliore di lavorare è definirli come una caratteristica o un'Epic.

A volte, inizialmente, a livello di release viene specificato "solo" un elenco di alto livello di attributi di qualità del prodotto. Questi saranno gradualmente elaborati durante le sessioni di perfezionamento come funzionalità, Epic o User Story. Successivamente, all'inizio di un'iterazione, il team valuterà se uno o più degli obiettivi di qualità del prodotto specificati faranno parte dell'obiettivo di quella specifica iterazione.

4.2.2 SG 2 I progressi effettivi verso il raggiungimento degli obiettivi di qualità del prodotto del progetto sono quantificati e gestiti

Una volta fissati gli obiettivi quantitativi di qualità del prodotto, questi saranno monitorati, controllati ed eventualmente adeguati se necessario. Poiché la qualità del prodotto sarà definita come parte del DoD e/o come caratteristiche specifiche, Epic o User Story con criteri di accettazione misurabili, per tracciare e gestire i progressi verso il raggiungimento degli obiettivi di qualità del prodotto si possono utilizzare gli stessi metodi e tecniche già spiegati e descritti nell'area di processo 2.3 Monitoraggio e controllo dei test al livello 2 di TMMi. I progressi saranno misurati quotidianamente e discussi nel corso della riunione giornaliera di standup che riporta i progressi, la qualità e i rischi. Inoltre, la valutazione della qualità del prodotto è tipicamente supportata anche dall'area di processo Test Measurement, che fornisce un'infrastruttura di misurazione.

4.3 Area di processo 4.3 Revisioni avanzate

Lo scopo delle revisioni avanzate, basate sulle pratiche dell'area di processo TMMi livello 3 delle Peer Review, è quello di misurare la qualità del prodotto nelle prime fasi del ciclo di vita e di migliorare la strategia e l'approccio di test allineando le Peer Review (test statici) con i test dinamici.

4.3.1 SG 1 Coordinare l'approccio di Peer Review con l'approccio di Test Dinamico

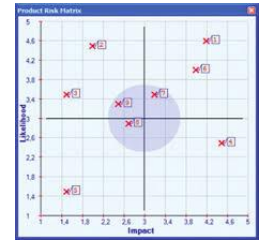
Mentre nei progetti tradizionali i test statici, ad esempio le revisioni, non sono spesso collegati ai test dinamici svolti da un team di test indipendente, in un contesto Agile la situazione dovrebbe essere completamente diversa. In un ambiente Agile, l'approccio per garantire la qualità del prodotto dovrebbe essere quello in cui tutte le attività di test, sia statiche che dinamiche, sono organizzate come un approccio coerente e coordinato. È ovviamente responsabilità del team Agile integrare i test statici e dinamici e trovare il mix ottimale. Il team Agile deve assumere un ruolo guida nello stabilire un approccio integrato coerente ed efficace. L'approccio per i test statici (peer review) sarà quindi allineato e coordinato con l'approccio per i test dinamici.

4.3.2 SG 2 Misurare la qualità del prodotto nelle prime fasi del ciclo di vita attraverso le peer review

Al livello 4 di TMMi, l'organizzazione stabilisce obiettivi quantitativi per i prodotti software e i relativi prodotti di lavoro. Le Peer Review svolgono un ruolo essenziale nel raggiungimento di questi obiettivi. Mentre al livello TMMi 3 le Peer Review vengono eseguite principalmente per individuare i difetti, ora l'enfasi è posta sulla misurazione della qualità del prodotto/documento per controllare la qualità del prodotto nelle prime fasi del ciclo di vita. Le pratiche di revisione sono migliorate per includere pratiche come il campionamento, l'applicazione di criteri di uscita e la prescrizione di regole. Questo è ciò che Tom Gilb chiama "ispezione agile": si tratta di spostare l'enfasi dalla pulizia al campionamento, alla misurazione, alla motivazione e alla prevenzione dei difetti [Gilb]. Questa è chiaramente una pratica che si adatta alle organizzazioni Agile ad alta maturità. Un altro esempio potrebbe essere quello di utilizzare le regole INVEST [Wake] per le User Story come Definition of Ready (DoR) e misurare la qualità delle User Story rivedendole rispetto a queste regole. Avere una definizione di Ready significa che le storie devono essere immediatamente attivabili.

4.3.3 SG 3 Adattare l'approccio di test in base ai risultati delle revisioni nelle prime fasi del ciclo di vita

Quando i risultati della peer review e i test dinamici sono coordinati, i risultati e i dati delle prime revisioni possono influenzare i rischi del prodotto e l'approccio ai test. Come già affermato nell'obiettivo specifico 1 di quest'area di processo, in Agile la qualità è un lavoro di team e i risultati delle attività di verifica e convalida vengono discussi durante le riunioni di team. Il principio dell'utilizzo dei primi risultati dei test per influenzare le attività di test successive si sposa quindi quasi perfettamente con il modo di lavorare dei team e dei progetti Agile.



5 Ottimizzazione di livello 5 di TMMi

Gli obiettivi e le pratiche del livello 5 di TMMi derivano da approcci di gestione della qualità e di miglioramento dei processi di elevata maturità. A questo livello di maturità gli obiettivi e le pratiche sono quasi indipendenti dal ciclo di vita del software che viene applicato. Naturalmente, un'organizzazione basata su Agile si approccerà in modo diverso rispetto alle organizzazioni tradizionali basate sulla gerarchia, ma le pratiche essenziali relative al controllo della qualità, all'ottimizzazione dei processi e alla prevenzione dei difetti rimarranno in gran parte invariate. Il motivo è che queste pratiche sono considerate pratiche di gestione della qualità, tipicamente eseguite a un livello superiore a quello dello sviluppo operativo del prodotto. Non si tratta quindi di pratiche direttamente collegate alle attività primarie di sviluppo e/o collaudo del software all'interno di un progetto o di un flusso di delivery del prodotto.

5.1 Area di processo 5.1 Prevenzione dei difetti

Lo scopo della prevenzione dei difetti è quello di identificare e analizzare le cause comuni dei difetti nel ciclo di vita dello sviluppo e definire le azioni per evitare che difetti simili si verifichino in futuro.

Le organizzazioni agili di solito si concentrano già sulla prevenzione dei difetti, che è una base importante su cui costruire gli obiettivi specifici e le pratiche specifiche di quest'area di processo. Le organizzazioni agili efficienti si concentrano sulla prevenzione dei difetti piuttosto che solo sulla loro scoperta. Una cultura della prevenzione dei difetti include la separazione dei difetti "in corso d'opera" da quelli che sfuggono all'iterazione in cui vengono sviluppate le User Story.

La retrospettiva agile è una pratica che consente ai team di riflettere, imparare e migliorare continuamente il proprio lavoro. Sebbene le retrospettive siano più spesso utilizzate per esplorare l'attuale modo di lavorare, possono anche essere usate per indagare su problemi di qualità o per concordare azioni che possano migliorare la qualità del software consegnato.

L'area di processo Prevenzione dei difetti si occupa delle pratiche per identificare e analizzare le cause comuni dei difetti tra i vari team, prodotti e flussi di valore, e definire azioni specifiche per eliminare le cause comuni di questi tipi di difetti in futuro. Tutti i difetti, siano essi riscontrati durante lo sviluppo, il collaudo o sul campo, rientrano nell'ambito di quest'area di processo. Poiché la prevenzione dei difetti a questo livello necessita di dati di misurazione, la prevenzione dei difetti si basa sulle pratiche di misurazione del livello 4 del TMMi e sui dati di misurazione disponibili relativi allo sviluppo, ai test e alla qualità del prodotto.

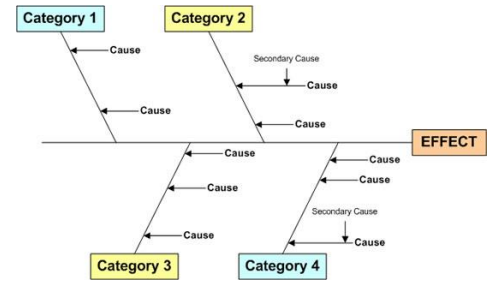
Nelle organizzazioni Agile esistono fondamentalmente due tipi distinti di attività di prevenzione dei difetti: a livello di team e a livello di cross-team. Le attività di prevenzione dei difetti a livello di team sono di responsabilità del team e sono già state trattate al livello 3 del TMMi, mentre le attività di prevenzione dei difetti a livello di cross-team sono svolte da un gruppo di processo di test o da una corporazione di test e sono quelle che vengono trattate in modo specifico da quest'area di processo.

5.1.1 SG 1 Determinare le cause comuni dei difetti

Nell'ambito di Agile i miglioramenti, compresi quelli del processo di test, avvengono generalmente sulla base di frequenti cicli di feedback. Poiché l'ambito di applicazione è spesso limitato all'iterazione precedente, vengono apportati miglioramenti piccoli ma frequenti che si concentrano principalmente sulla risoluzione di specifici problemi locali. Spesso questi miglioramenti non si concentrano sull'apprendimento trasversale e sull'istituzionalizzazione dei miglioramenti. Risolvere i problemi solo a livello di team può anche portare facilmente alla sub-ottimizzazione e alla perdita di contatto con il quadro generale. È qui che quest'area di processo TMMi aggiunge valore: basandosi sulle pratiche Agile esistenti, la prevenzione dei difetti viene ora applicata anche a livello organizzativo (tra i team, i prodotti e i flussi di valore) e tipicamente gestita da un gruppo di processo (di test) o da una corporazione di test (vedere paragrafo 3.1.1).

La selezione dei difetti da analizzare deve basarsi su vari fattori, tra cui il rischio e il valore aziendale. È necessario concentrarsi sulle aree in cui la prevenzione dei difetti ha il maggior valore aggiunto (di solito in termini di riduzione dei costi o dei rischi) e/o in cui i difetti sono più critici. Sebbene l'area del processo di prevenzione dei difetti sia comunemente utilizzata per i difetti, può essere utilizzata anche per problemi quali la velocità.

Anche se un gruppo di processo di test può coordinare le attività di prevenzione dei difetti, questo dovrebbe essere fatto da un team multidisciplinare, ad esempio con rappresentanti dell'ingegneria dei requisiti, dell'ingegneria di sistema e/o dello sviluppo del software, poiché le azioni di miglioramento spesso interessano altre discipline. Naturalmente, questo è già comune nei metodi di delivery e nei team Agile. Assicuratevi comunque che tutte le discipline richieste siano coinvolte, come per le azioni di miglioramento. I metodi analitici per valutare le cause alla radice dei difetti, come i diagrammi causa-effetto, i diagrammi a lisca di pesce o i 5 perché, sono tipicamente utilizzati per la prevenzione dei difetti e sono già pratiche Agile comuni.



Tutte e tre le pratiche specifiche (1.1 Definire i parametri di selezione dei difetti, 1.2 Selezionare i difetti per l'analisi e 1.3 Analizzare le cause principali e comuni dei difetti selezionati) definite dal modello TMMi per questo obiettivo specifico sono applicabili anche a un'organizzazione Agile.

5.1.2 SG 2 Prioritizzare e definire le azioni per eliminare sistematicamente le cause comuni dei difetti

Non ci sono differenze reali nell'esecuzione delle pratiche specifiche nell'ambito di questo obiettivo, sia che vengano praticate in un ciclo di vita Agile che sequenziale. Naturalmente la soluzione effettivamente proposta e la successiva proposta di miglioramento saranno diverse, in quanto devono adattarsi al modo di lavorare Agile, ad esempio creando voci di backlog aggiuntive invece di proposte di azione formalizzate con approvazioni. Vengono intraprese azioni appropriate per ridurre al minimo la probabilità che il problema si ripresenti. Spesso questo significa combinare il miglioramento dei processi con un tutoraggio e/o una formazione aggiuntiva per i membri del team e altro team. È necessario affrontare sia i problemi delle persone che quelli dei processi in modo integrato, piuttosto che cercare di separarli artificialmente.

Inoltre, le organizzazioni Agile si basano su team che si autogestiscono. Ciò implica che i team possono scegliere quali miglioramenti implementare e/o rifiutare. Per questo motivo, durante la definizione delle priorità e delle azioni, è necessario coinvolgere una rappresentanza adeguata dei team per discutere se questi miglioramenti forniscono effettivamente un valore aggiunto per loro.

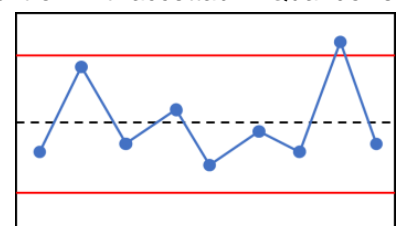
Anche il modo in cui i miglioramenti saranno distribuiti sarà diverso, meno dall'alto verso il basso e più simile a un'offerta su base volontaria ai team che si autogestiscono. Per saperne di più sull'implementazione dei miglioramenti dei test, si veda l'area di processo Test Process Optimization (Ottimizzazione del processo di test).

5.2 Area di processo 5.2 Controllo qualità

Lo scopo del Controllo qualità è quello di gestire e controllare statisticamente il processo di test. A questo livello, le prestazioni del processo di test sono completamente prevedibili e stabilizzate entro limiti accettabili. I test a livello di progetto vengono eseguiti utilizzando metodi statistici basati su campioni rappresentativi, al fine di prevedere la qualità del prodotto e rendere i test più efficienti.

5.2.1 SG 1 Stabilire un processo di test statisticamente controllato

I principi e gli elementi essenziali del controllo di qualità sono gli stessi indipendentemente dal metodo del ciclo di vita applicato. Il controllo della qualità del processo prevede la definizione di obiettivi per le prestazioni del processo di test standard. Sulla base delle misurazioni effettuate dai team sulle prestazioni del processo di test, si procede all'analisi e alle regolazioni per mantenere le prestazioni del processo di test entro limiti accettabili. Quando le prestazioni del processo di test si stabilizzano entro limiti accettabili, il processo di test definito, le misurazioni associate e i limiti accettabili per le misurazioni vengono stabiliti come linea di base e utilizzati per controllare statisticamente le prestazioni del processo di test. La capacità del processo di test standard dell'organizzazione, cioè le prestazioni del processo di test che un team può aspettarsi di ottenere, è ora pienamente compresa e conosciuta. Di conseguenza,



le deviazioni da queste aspettative possono essere affrontate dal team in modo tempestivo e coerente, per garantire prestazioni entro i limiti accettabili.

È importante selezionare i sottoprocessi per il controllo statistico che sono critici per l'azienda e che hanno avuto problemi in passato. L'esperienza indica che il più delle volte i sottoprocessi che meglio soddisfano questi criteri non sono isolati a singoli team, ma piuttosto attraversano più team, progetti e flussi di consegna del prodotto. Quando si utilizzano metodi e tecniche Agile, il lavoro viene suddiviso in brevi iterazioni, in cui il lavoro completato in ogni iterazione comporta l'esecuzione di molteplici attività di sviluppo e di test. Di conseguenza, lo sviluppo Agile supporta, in modo più efficace rispetto allo sviluppo tradizionale, il monitoraggio dei sottoprocessi più importanti per il controllo statistico, poiché i cicli sono molto più brevi e consentono un feedback quasi istantaneo sulle prestazioni del processo di test. La rapidità del feedback è tipicamente molto preziosa per i team ed è un importante principio di Agile.

5.2.2 SG 2 Il test è eseguito con metodi statistici

Il controllo della qualità del prodotto si basa sui profili operativi [Musa] e sui modelli di utilizzo del prodotto nell'ambiente in cui è destinato, per fare inferenze statisticamente valide e ottenere un campione rappresentativo di test. Questo approccio, particolarmente utile a livello di sistema, utilizza metodi di test statistici per prevedere la qualità del prodotto sulla base di questo campione rappresentativo. In altre parole, quando si testa un sottoinsieme di tutti i possibili utilizzi rappresentati dal profilo d'uso o operativo, i risultati dei test possono servire come base per le conclusioni sulle prestazioni complessive del prodotto.

Si tratta di un approccio che si adatta bene ad Agile, perché è snello per natura ed è guidato dal feedback dell'uso operativo e dalle sessioni di revisione del prodotto. Tuttavia, richiede che un'organizzazione Agile abbia il giusto livello di maturità per poter applicare questo approccio. Inoltre, si basa e dipende fortemente dalle misurazioni e dai dati raccolti e disponibili.

Un'organizzazione che utilizza metodi statistici nei test è in grado di quantificare i livelli di confidenza e di affidabilità (per maggiori informazioni su questi termini, consultare il framework TMMi). Sia il livello di fiducia che l'attendibilità sono tipicamente utilizzati come Definition of Done quando si applicano i test statistici.

5.3 Area di processo 5.3 Ottimizzazione del processo di test

Lo scopo dell'ottimizzazione del processo di test è quello di migliorare continuamente i processi di test esistenti utilizzati nell'organizzazione e di identificare nuove tecnologie di test (ad esempio, strumenti o metodi di test) che potrebbero essere appropriate e di introdurle nell'organizzazione in modo ordinato. L'ottimizzazione dei processi di test supporta anche il riutilizzo delle risorse di test all'interno dell'organizzazione. I miglioramenti supportano gli obiettivi di qualità del prodotto e di performance del processo di test dell'organizzazione, derivati dagli obiettivi aziendali.

5.3.1 SG 1 Selezionare i miglioramenti del processo di test

Vengono raccolte e analizzate le proposte di miglioramento dei test. Le proposte di miglioramento dei test provengono da fonti interne e sono tipicamente cross-team, cross-project e cross-product stream. Le fonti interne includono problemi e idee provenienti da attività di prevenzione dei difetti, revisioni del team, retrospettive, capitoli e corporazioni. Dopo un'analisi dei costi e dei benefici, le proposte interessanti di miglioramento dei test vengono sperimentate per valutare i cambiamenti più importanti, nuovi e non provati, prima di essere distribuiti a livello organizzativo nei team, a seconda dei casi. Infine, viene effettuata una selezione dei miglioramenti del test da distribuire all'interno dell'organizzazione.

5.3.2 SG 2 Le nuove tecnologie di test vengono valutate per determinarne l'impatto sul processo di test.

Nell'ambito dell'ottimizzazione del processo di test, il gruppo di processo di test organizzato centralmente (o squadra di test) ricerca proattivamente sul mercato nuove tecnologie, come strumenti, metodi, tecniche o innovazioni tecniche, che possano migliorare la capacità di test dei team Agile. Mantenendo una consapevolezza delle innovazioni tecnologiche legate al test e valutandole e sperimentandole sistematicamente, l'organizzazione, in consultazione con i rappresentanti dei team, seleziona le tecnologie di test appropriate per migliorare la qualità dei

prodotti e la produttività delle attività di test. All'interno dei team vengono eseguiti progetti pilota per valutare le tecnologie di test nuove e non collaudate prima che vengano incorporate nella pratica standard.

5.3.3 SG 3 Distribuzione dei miglioramenti del test

I miglioramenti dei test e le nuove tecnologie di test appropriate vengono distribuiti tra i team Agile per migliorare i test. I loro benefici vengono misurati e le informazioni sulle nuove innovazioni vengono diffuse in tutta l'organizzazione. Esiste naturalmente una differenza tra l'implementazione in un'organizzazione tradizionale e quella in un'organizzazione Agile. In un'organizzazione tradizionale, il più delle volte il deployment si traduce in nuove pratiche che devono essere applicate da tutti i soggetti coinvolti. In un'organizzazione Agile il team ha maggiore autonomia e si autogestisce. Questo implica anche che il processo di distribuzione è diverso. I miglioramenti dei test e le nuove tecnologie di test appropriate vengono incorporati nel processo di test dell'organizzazione e, se necessario, vengono organizzati e offerti formazione e supporto. Tuttavia, alla fine è il team Agile stesso a decidere se il miglioramento del test diventerà o meno parte del suo modo di lavorare.

5.3.4 SG 4 Stabilire il riutilizzo di risorse di test di alta qualità

Al livello 3 di TMMi può già avvenire un certo riutilizzo del testware tra team, progetti e flussi di prodotto; tuttavia, il riutilizzo delle risorse di test diventa un obiettivo importante al livello 5 di TMMi. Il team non deve reinventare la ruota, ma piuttosto attingere alle competenze e alle risorse già disponibili, risparmiando così fatica e tempo. Vengono identificati gli asset di test di alta qualità che possono essere riutilizzati in tutta l'organizzazione, ad esempio durante le riunioni retrospettive, le sessioni di lesson learned, le discussioni all'interno del gruppo di test o le valutazioni dei test. Successivamente, gli asset di test riutilizzabili vengono selezionati e aggiunti al repository centrale di test in un formato modificabile per il riutilizzo in progetti e delivery di prodotti futuri.

6 Panoramica Applicabilità Obiettivi e pratiche specifiche TMMi

6.1 Valutazioni TMMi

Per aiutare i lead-assessor e i valutatori nelle loro valutazioni, in questo capitolo viene fornita una panoramica dell'applicabilità delle aree di processo, degli obiettivi specifici e delle pratiche specifiche in un contesto Agile. Nei capitoli precedenti sono state identificate alcune pratiche specifiche che si prevede siano meno rilevanti o forse addirittura non rilevanti. Queste pratiche specifiche sono elencate di seguito. Durante una valutazione TMMi, un lead-assessor o un valutatore dovrà valutare esplicitamente questi item per la loro rilevanza nello specifico contesto Agile. Nel caso in cui si decida che un elemento è effettivamente meno rilevante o addirittura non rilevante, questo dovrebbe essere chiaramente documentato con una motivazione, sia nei processi dell'organizzazione, che nel rapporto di valutazione.

6.2 TMMi livello 2 Gestito

Area di processo 2.2 Pianificazione dei test

SG2 Sviluppare un approccio di test	Applicabile
- SP2.3 Definire i criteri di ingresso	<i>Probabilmente meno/ non rilevante</i>
- SP2.5 Definire criteri di sospensione e di ripresa	<i>Probabilmente meno/ non rilevante</i>
SG3 Stabilire le stime dei test	Applicabile
- S3.2 Definire il ciclo di vita dei test	<i>Probabilmente meno/non rilevante</i>
SG5 Ottenere l'impegno a rispettare il Test Plan	Applicabile
- SP5.2 Riconciliare i livelli di lavoro e di risorse	<i>Probabilmente meno/non rilevante</i>

Area di processo 2.3 Monitoraggio e controllo dei test

SG2 Monitoraggio della qualità del prodotto rispetto al piano e alle aspettative	Applicabile
- SP2.3 Monitoraggio dei criteri di ingresso	<i>Probabilmente meno/non rilevante</i>
- SP2.5 Monitoraggio dei criteri di sospensione e di ripresa	<i>Probabilmente meno/non rilevante</i>

Area di processo 2.4 Progettazione ed esecuzione dei test

SG2 Esecuzione del test di implementazione	Applicabile
- SP2.3 Specificare il test di assunzione	<i>Probabilmente meno/ non rilevante</i>
- SP2.4 Sviluppare il programma di esecuzione dei test	<i>Probabilmente meno/ non rilevante</i>
SG3 Esecuzione del test	Applicabile
- SP3.1 Eseguire il test di ingresso	<i>Probabilmente meno/ non rilevante</i>

6.3 TMMi livello 3 Definito

Area di processo 3.5 Peer Review

SG2 Eseguire Peer Review	Applicabile
- SP2.3 Analizzare i dati della peer review	<i>Probabilmente meno/ non rilevante</i>

6.4 TMMi livello 4 Misurato

Tutte le aree di processo, gli obiettivi specifici e le pratiche specifiche del TMMi livello 4 sono applicabili. Nessuna pratica specifica è indicata come probabilmente meno/ non rilevante.

6.5 Ottimizzazione TMMi livello 5

Tutte le aree di processo, gli obiettivi specifici e le pratiche specifiche del TMMi livello 5 sono applicabili. Nessuna pratica specifica è indicata come probabilmente meno/ non rilevante.

Riferimenti

- [Black, Van Veenendaal] Rex Black, Erik van Veenendaal e Dorothy Graham (2012), *Foundations of Software Testing ISTQB Certification* (3rd edizione), Cengage
- [Cohn] M. Cohn (2009), *Succeeding with Agile: Software Development using Scrum*, Addison-Wesley
- [Galen] R. Galen (2015), *Three Pillars of Agile Quality and Testing: Ottenere risultati equilibrati nel viaggio verso la qualità agile*, RGCG, LLC
- [T. Gilb (2004), *Agile Specification Quality Control*, International Council on Systems Engineering (INCOSE).
- [ISO/IEC 25010 (2011), *Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) System and Software Quality Models*, Organizzazione internazionale per la standardizzazione.
- [Sutherland et al] C.R. Jakobsen, K. Johnson e J. Sutherland (2007), *Scrum e CMMI livello 5: la porzione magica per i guerrieri del codice* in: Conferenza IEEE Agile 2007, Washington, D.C.
- [McMahon] Paul. E. McMahon (2011), *Integrating CMMI with Agile Development - Case Studies and Proven Techniques for Faster Performance Improvement*, Addison Wesley.
- [Musa] J. Musa (1998), *Software Reliability Engineering Testing*, McGraw-Hill Education
- [Van der Aalst e Davis] L. van der Aalst e Cecile Davis (2013), *TMap NEXT in scrum - Effective testing in agile projects*, Sogeti Nederland B.V.
- [Van Solingen e Berghout] R. van Solingen e E. Berghout (1999), *The Goal/Question/Metric method*, McGrawHill
- [van Veenendaal] E. van Veenendaal (2014), *PRISMA: Product Risk Assessment for Agile Projects*, in: *Testing Experience*, numero 04/12, dicembre 2012.
- [Wake] B. Wake (2003), *INVEST in Good Stories, and SMART Tasks*, xp123.com/articles/invest-in-good-stories-and-smart-tasks