

Certified Tester Advanced Level Test Management Syllabus

Versione 3.0

International Software Testing Qualifications Board



Avviso di copyright

Avviso di Copyright © International Software Testing Qualifications Board (di seguito denominato ISTQB®)
ISTQB® è un marchio registrato di International Software Testing Qualifications Board.

Copyright © 2023, gli autori dell'aggiornamento 3.0 sono: Horst Pohlmann (Product Owner, Vice Chair AELWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma e Miroslav Renda.

Copyright © 2010-2012 gli autori per l'Advanced Level Test Manager Sub Working Group: Rex Black (Presidente), Judy McKay (Vicepresidente), Graham Bath, Debra Friedenber, Bernard Homès, Kenji Onishi, Mike Smith, Geoff Thompson, Tsuyoshi Yumoto.

Tutti i diritti riservati. Con la presente gli autori trasferiscono il copyright a ISTQB®. Gli autori (come attuali titolari del copyright) e ISTQB® (come futuro titolare del copyright) hanno concordato le seguenti condizioni di utilizzo:

- È possibile copiare estratti di questo documento per uso non-commerciale, a condizione che venga riconosciuta la fonte. Qualsiasi Training Provider accreditato (azienda accreditata alla Formazione) può utilizzare il presente Syllabus come base per un corso di formazione, a condizione che gli autori e ISTQB® siano riconosciuti come fonte e proprietari del copyright del Syllabus, e a condizione che qualsiasi pubblicità di tale corso di formazione possa menzionare il Syllabus solo dopo l'accREDITAMENTO ufficiale dei materiali di formazione da parte di un Member Board riconosciuto da ISTQB®.
- Qualsiasi individuo o gruppo di individui può utilizzare questo Syllabus come base per articoli e libri, a condizione che gli autori e ISTQB® siano riconosciuti come fonte e proprietari del copyright del Syllabus.
- È proibito qualsiasi altro utilizzo di questo Syllabus senza avere prima ottenuto l'approvazione scritta di ISTQB®.
- Qualsiasi Member Board riconosciuto da ISTQB® può tradurre questo Syllabus, a condizione che riproduca il suddetto avviso di copyright nella versione tradotta del Syllabus.

Storico delle review

Versione	Data	Osservazioni
ISEB v1.1	04.09.2001	Syllabus ISEB Practitioner
ISTQB 1.2E	09/2003	Syllabus ISTQB Advanced Level da EOQ-SG
V2007	12.10.2007	Syllabus Certified Tester Advanced Level versione 2007
D100626	10.06.2010	Inclusione delle modifiche accettate nel 2009, separazione di ogni capitolo per i moduli separati
D101227	10.12.2010	Accettazione delle modifiche al formato, e correzioni che non hanno alcun impatto sul significato delle frasi.
D2011	13.10.2011	Modifica in un Syllabus dedicato, rielaborazione dei LO e modifica del testo per adeguarlo ai LO. Aggiunta di BO.
Alpha 2012	09.02.2012	Inclusione di tutti i commenti di NB ricevuti a partire dalla pubblicazione di ottobre.
Beta 2012	23.06.2012	Inclusione dei commenti di NB ricevuti dall'Alpha release.
Beta 2012	07.04.2012	Versione beta sottoposta a GA
Beta 2012	08.06.2012	Versione modificata rilasciata a NB
Beta 2012	27.06.2012	Commenti di EWG e Glossario incorporati
RC 2012	18.05.2012	Versione candidata al rilascio - incluse le modifiche finali di NB
Beta v3.0	31.10.2023	Inclusione di tutti i commenti dei member board ricevuti per tutte le sezioni (BOIncs) durante l'Alpha release
POST Beta v3.0	13.01.2024	Inclusione di tutti i commenti dei member board ricevuti per tutte le sezioni (BOIncs) durante la Beta release
POST Beta v3.0	29.02.2024	Modifiche minori nella correzione di bozze
RC v3.0	28.03.2024	Versione candidata al rilascio – inclusione delle ultime modifiche al modello formale come proposto da PWG
V.3.0	03.05.2024	Rielaborazione dopo il rilascio; eliminati solo gli errori di battitura e le incongruenze

Indice dei contenuti

Avviso di copyright	2
Storico delle review	3
Indice dei contenuti	4
Ringraziamenti	7
0 Introduzione	9
0.1 Scopo di questo Syllabus	9
0.2 Il Certified Tester Advanced Level Test Management nel Testing	9
0.3 Percorso di Carriera per i Tester	9
0.4 Business Outcome	10
0.5 Obiettivi di Apprendimento Esaminabili e Livelli di Conoscenza	10
0.6 L'Esame per il Certificato Advanced Level Test Management	11
0.7 Accredimento	11
0.8 Gestione degli Standard	11
0.9 Livello di Dettaglio	12
0.10 Come è Organizzato questo Syllabus	12
0.11 Quali sono le Assunzioni Fondamentali di questo Syllabus?	13
1 Gestire le Attività di Test - 750 minuti	15
1.1 Il Processo di Test	17
1.1.1 Attività di Pianificazione dei Test	17
1.1.2 Attività di Monitoraggio e Controllo dei Test	18
1.1.3 Attività di Completamento dei Test	19
1.2 Il Contesto del Testing	20
1.2.1 Stakeholder del Test	20
1.2.2 Importanza della Conoscenza degli Stakeholder nel Test Management	20
1.2.3 Il Test Management in un Modello di Sviluppo Software Ibrido	22
1.2.4 Le Attività di Test Management per i Diversi Modelli del Ciclo di Vita dello Sviluppo Software	22
1.2.5 Attività di Test Management nei Diversi Livelli di Test	23
1.2.6 Attività di Test Management per Diversi Tipi di Test	24
1.2.7 Attività di Test Management per Pianificare, Monitorare e Controllare	25
1.3 Testing Basato sul Rischio	27

1.3.1	Il Testing come Attività di Mitigazione del Rischio	27
1.3.2	Identificazione dei Rischi di Qualità	27
1.3.3	Valutazione del Rischio di Qualità	28
1.3.4	Mitigazione del Rischio di Qualità attraverso un Testing Appropriato	29
1.4	La Strategia di Test del Progetto	33
1.4.1	Scegliere un Approccio del Test	33
1.4.2	Analizzare la Strategia di Test dell'Organizzazione, il Contesto di Progetto e Altri Aspetti	34
1.4.3	Definizione degli Obiettivi del Test	35
1.5	Migliorare il Processo di Test	37
1.5.1	Il Processo di Miglioramento del Test (IDEAL)	37
1.5.2	Test Process Improvement Model-based	38
1.5.3	Approccio al Test Process Improvement Analitico	39
1.5.4	Retrospective	40
1.6	Strumenti di Test	42
1.6.1	Buone Pratiche per l'Introduzione di uno Strumento	42
1.6.2	Aspetti Tecnici e di Business per le Decisioni sugli Strumenti	43
1.6.3	Considerazioni sul Processo di Selezione e Valutazione del Return on Investment (ROI, Ritorno sull'Investimento)	43
1.6.4	Ciclo di Vita dello Strumento	45
1.6.5	Metriche degli Strumenti	45
2	Gestire il Prodotto - 390 minuti	47
2.1	Metriche di Test	48
2.1.1	Metriche per le Attività di Test Management	48
2.1.2	Monitoraggio, Controllo e Completamento	49
2.1.3	Reporting del Test	50
2.2	Stima del Test	52
2.2.1	Stimare quali Attività Coinvolgeranno il Testing	52
2.2.2	Fattori che Possono Influenzare l'Effort del Test	52
2.2.3	Selezione delle Tecniche di Stima del Test	53
2.3	Defect Management	56
2.3.1	Ciclo di Vita del Difetto	56
2.3.2	Defect Management Cross-funzionale	58
2.3.3	Specifiche del Defect Management nei Team Agile	59

2.3.4 Sfide del Defect Management nello Sviluppo Software Ibrido	59
2.3.5 Informazioni del Defect Report.....	60
2.3.6 Definire le Azioni di Process Improvement Utilizzando le Informazioni del Defect Report	61
3 Gestire il Team - 225 minuti.....	63
3.1 Il Team di Test	64
3.1.1 Competenze Tipiche all'Interno delle Quattro Aree di Competenza	64
3.1.2 Analizzare le Competenze Richieste ai Membri del Team di Test.....	65
3.1.3 Valutare le Competenze dei Membri del Team di Test.....	66
3.1.4 Sviluppare le Competenze dei Membri del Team di Test	67
3.1.5 Competenze di Management Necessarie per Gestire un Team di Test	67
3.1.6 Fattori Motivanti o Demotivanti per un Team di Test in Determinate Situazioni	68
3.2 Relazioni con gli Stakeholder.....	70
3.2.1 Costo della Qualità	70
3.2.2 Relazione Costi-benefici del Testing	70
4 Riferimenti	73
Standard	73
Documenti ISTQB®	73
Libri	73
Articoli	74
Pagine Web	74
5 Appendice A - Obiettivi di Apprendimento/Livello Cognitivo di Conoscenza.....	75
Livello 1: Ricordare (K1)	75
Livello 2: Comprendere (K2).....	75
Livello 3: Applicare (K3).....	76
Livello 4: Analizzare (K4)	76
6 Appendice B - Matrice di Tracciabilità dei Business Outcome rispetto agli Obiettivi di Apprendimento	78
7 Appendice C - Release Note	87
8 Appendice D - Parole Chiave Specifiche del Dominio.....	89
9 Appendice E – Marchi Registrati.....	90
10 Indice.....	91

Ringraziamenti

Questo documento è stato formalmente rilasciato dalla General Assembly di ISTQB® il 3 maggio 2024.

È stato prodotto da un team di International Software Testing Qualifications Board: Horst Pohlmann (Product Owner, Vice Chair AELWG), Tauhida Parveen, Francis Fenner, Laura Albert, Matthias Hamburg, Maud Schlich, Tanja Tremmel, Ralf Bongard, Erik van Veenendaal, Jan Giessen, Bernd Freimut, Andreas Neumeister, Georg Sehl, Rabi Arabi, Therese Kuhfuß, Ecaterina Irina Manole, Veronica Belcher, Kenji Onishi, Pushparajan Balasubramanian, Meile Postuma e Miroslav Renda.

Il team ringrazia Gary Mogyorodi per la sua review tecnica (versione Beta), Julia Sabatine per la sua correzione delle bozze, il gruppo di review e i Member Board per i loro suggerimenti e input forniti.

Hanno partecipato alla review, ai commenti e alla votazione di questo Syllabus:

Review Alpha: Benjamin Timmermans, Mattijs Kemmink, Rik Marselis, Jean-Francois Riverin, Gary Mogyorodi, Ralf Bongard, Ingvar Nordström, Yaron Tsubery, Imre Mészáros, Mattijs Kemmink, Ádám Bíró, Ramit M Kaul, Chinthaka Indikadahena, Darvay Tamás Béla, Beata Karpinska, Young jae Choi, Stuart Reid, Tal Pe'er, Meile Postuma, Daniel van der Zwan, Klaudia Dussa-Zieger, Jörn Münzel, Ralf Bongard, Petr Neugebauer, Derk-Jan de Groot, Rik Kochuyt, Andreas Hetz, Laura Albert, Eszter Sebestyeni, Tamás Szőke, Henriett Braunné Bokor, Ágota Horváth, Péter Sótér, Ferenc Hamori, Darvay Tamás Béla, Paul Weymouth, Lloyd Roden, Kevin Chen, Huang qin, Pushparajan Balasubramanian, Szilard Szell, Tamas Stöckert, Lucjan Stapp, Adam Roman, Anna Miazek, Márton Siska, Erhardt Wunderlich, László Kvintovics, Murian Song, Mette Bruhn-Pedersen, Petra Schneider, Michael Stahl, Ramit M Kaul, Imre Mészáros, Dilhan Jayakody, Francisca Cano Ortiz, Johan Klinton, Liang Ren, Ole Chr. Hansen, Zsolt Hargitai, Tamás Rakamazi, Kenji Onishi, Arnika Hryszsko, Rabih Arabi, Veronica Belcher e Vignesh Balasubramanian.

Review Beta: Maria-Therese Teichmann, Dominik Weber, Thomas Puffler, Peter Kunit, Martin Klönk, Michaël Pilaeten, Wim Decoutere, Arda Ender Torçuk, Piet de Roo, Rik Marselis, Jakub Platek, Ding Guofu, Zheng Dandan, Liang Ren, Yifan Chen, Hallur Helmsdal, Ole Chr. Hansen, Klaus Skafte, Gitte Ottosen, Tanzeela Gulzar, Arne Becher, Klaudia Dussa-Zieger, Jan Giesen, Florian Fieber, Carsten Weise, Arnd Pehl, Matthias Hamburg, Stephanie Ulrich, Jürgen Beniermann, Márton Siska, Sterbinszky Ádám, Ágnes Srancsik, Marton Matyas, Tamas Stöckert, Csilla Varga, Zsolt Hargitai, Bíró Ádám, Horváth Ágota, Sebestyeni Eszter, Szilárd Széll, Péter Sótér, Giancarlo Tomasig, Nicola de Rosa, Kaiwalya Katyarmal, Pradeep Tiwari, Sreeja Padmakumari, Seunghee Choi, Stuart Reid, Dmitrij Nikolajev, Mantas Aniulis, Monika Stoecklein-Olsen, Adam Roman, Mahmoud Khalaili, Ingvar Nordström, Beata Karpinska, Armin Born, Ferdinand Gramsamer, Mergole Kuate, Thomas Letzkus, Nishan Portoyan, Ainsley Rood, Lloyd Roden, Sarah Ireton.

Il Syllabus Advanced Level Test Manager 2010-2012 è stato prodotto da un core team dell'International Software Testing Qualifications Board Advanced Level Sub Working Group - Advanced Test Manager: Rex Black (Presidente), Judy McKay (Vicepresidente), Graham Bath, Debra Friedenberg, Bernard Homès, Paul Jorgensen, Kenji Onishi, Mike Smith, Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

Il core team ringrazia il gruppo di review e i National Board per i loro suggerimenti e input.

Al momento del completamento del Syllabus Advanced Level, l'Advanced Level Working Group era composto dai seguenti membri (in ordine alfabetico):

Graham Bath, Rex Black, Maria Clara Choucair, Debra Friedenberg, Bernard Homès (Vicepresidente), Paul Jorgensen, Judy McKay, Jamie Mitchell, Thomas Mueller, Klaus Olsen, Kenji Onishi, Meile Postuma, Eric Riou du Cosquer, Jan Sabak, Hans Schaefer, Mike Smith (Presidente), Geoff Thompson, Erik van Veenendaal, Tsuyoshi Yumoto.

Hanno partecipato alla review, ai commenti e alla votazione di questo Syllabus:

Chris van Bael, Graham Bath, Kimmo Hakala, Rob Hendriks, Marcel Kwakernaak, Rik Marselis, Don Mills, Gary Mogyorodi, Thomas Mueller, Ingvar Nordstrom, Katja Piroué, Miele Posthuma, Nathalie Rooseboom de Vries, Geoff Thompson, Jamil Wahbeh, Hans Weiberg.

Traduzione italiana a cura di Raffaele Bozzini, Salvatore Reale, Cristina Sobrero.

0 Introduzione

0.1 Scopo di questo Syllabus

Questo Syllabus costituisce la base per l'International Software Testing Qualification a livello Advanced per il Test Management. ISTQB® fornisce questo Syllabus come segue:

1. Ai Member Board, per la traduzione nella loro lingua locale e per l'accreditamento dei Training Provider. I Member Board possono adattare il Syllabus alle loro specifiche esigenze linguistiche e modificare i riferimenti per adattarli alle loro pubblicazioni locali.
2. Agli organismi di certificazione, per derivare le domande d'esame nella loro lingua locale adatte agli obiettivi di apprendimento di questo Syllabus.
3. Ai Training Provider accreditati ISTQB®, per produrre materiale didattico e determinare i metodi di insegnamento appropriati.
4. Ai candidati alla certificazione, per prepararsi all'esame di certificazione (ISTQB® raccomanda di seguire un corso di formazione accreditato da ISTQB® prima di sostenere un esame ISTQB® Advanced Level).
5. Alla comunità internazionale di System Engineering e Software Engineering, per promuovere la professione del testing dei sistemi e del software, e come base per libri e articoli.

0.2 Il Certified Tester Advanced Level Test Management nel Testing

La certificazione Advanced Level è rivolta a chiunque sia coinvolto nella gestione del testing del software. Questo include persone che ricoprono ruoli come tester, consulenti del test, test manager, user acceptance tester, scrum master, project manager o product owner. Questa certificazione Advanced Level Test Management è appropriata anche a chiunque desideri una comprensione avanzata del testing del software, come project manager, quality manager, responsabili dello sviluppo software, business analyst, direttori IT e consulenti manageriali.

I possessori della certificazione Advanced Level Test Management saranno in grado di accedere alle certificazioni del testing del software ISTQB® Expert Level. Il certificato ISTQB® Certified Tester Advanced Level - Test Manager o Test Management è valido a vita e non richiede rinnovo. Il certificato è riconosciuto a livello internazionale e dimostra la competenza professionale e la credibilità dei candidati nel test management.

0.3 Percorso di Carriera per i Tester

Lo schema ISTQB® fornisce supporto ai professionisti del testing in tutte le fasi della loro carriera.

Chi ha conseguito la certificazione ISTQB® Certified Tester Advanced Level Test Management può essere interessato anche alle altre certificazioni Core Advanced Level (cioè Test Analyst e Technical Test Analyst) e successivamente alle certificazioni ISTQB® Expert Level (cioè Test Management o Migliorare il Processo di Test).

Chiunque voglia sviluppare competenze relative alle pratiche del testing, nell'ambito dello sviluppo software Agile, può prendere in considerazione le certificazioni Agile Technical Tester o Agile Test Leadership at Scale.

Il percorso di certificazione Specialist offre un approfondimento in aree che hanno specifici approcci di test e attività di test (p.e. test automation, testing dell'intelligenza artificiale o testing di applicazioni mobile), o che raggruppano il know-how del testing per determinati domini industriali (p.e. automotive o gaming).

È possibile visitare il sito www.istqb.org per rimanere aggiornati sulle ultime informazioni relative allo schema di certificazione ISTQB®.

0.4 Business Outcome

Questo paragrafo elenca gli undici Business Outcome (Risultati di Business) previsti per una persona che ha conseguito la certificazione Advanced Level Test Management.

Un tester con la certificazione Advanced Level Test Management è in grado di...

TM_01	Gestire il testing nei vari progetti di sviluppo software applicando i processi di test management stabiliti per il team di progetto o per l'organizzazione del test
TM_02	Identificare gli stakeholder del test e i modelli del ciclo di vita dello sviluppo software che sono rilevanti in un determinato contesto
TM_03	Organizzare sessioni di identificazione del rischio e di valutazione del rischio all'interno di qualsiasi ciclo di vita dello sviluppo software, e utilizzare i risultati per guidare il testing per raggiungere gli obiettivi del test
TM_04	Definire una strategia di test del progetto consistente con la strategia di test dell'organizzazione e con il contesto di progetto
TM_05	Monitorare e controllare continuamente il testing per raggiungere gli obiettivi del progetto
TM_06	Valutare e fornire report sull'avanzamento del test agli stakeholder di progetto
TM_07	Identificare le competenze necessarie e sviluppare queste competenze all'interno del proprio team
TM_08	Preparare e presentare un business case per il testing in diversi contesti che delinei i costi e i benefici attesi
TM_09	Guidare le attività di test process improvement (miglioramento del processo di test) nei progetti o nei cicli di vita dello sviluppo software di un prodotto, e contribuire alle iniziative di test process improvement dell'organizzazione
TM_10	Pianificare le attività di test includendo l'infrastruttura del test necessaria, e stimare l'effort richiesto per il testing
TM_11	Creare i defect report e un workflow del difetto che siano adeguati al ciclo di vita dello sviluppo software

0.5 Obiettivi di Apprendimento Esaminabili e Livelli di Conoscenza

Gli Obiettivi di Apprendimento (LO, Learning Objective) supportano i Business Outcome e vengono utilizzati per creare gli esami di Certified Tester Advanced Level Test Management.

In generale, tutti i contenuti di questo Syllabus sono esaminabili a livello K1, ad eccezione di Introduzione, Riferimenti, Epilogo e Appendici. Al candidato può quindi essere chiesto di riconoscere, ricordare o richiamare una parola chiave o un concetto menzionato in uno qualsiasi dei tre capitoli di questo Syllabus.

I livelli specifici degli obiettivi di apprendimento sono specificati all'inizio di ogni capitolo e sono classificati come segue:

- K2: Capire
- K3: Applicare
- K4: Analizzare

Ulteriori dettagli ed esempi di obiettivi di apprendimento sono riportati nell'Appendice A.

Tutti i termini elencati come parole chiave sotto i titoli dei capitoli devono essere ricordati (K1), anche se non esplicitamente menzionati negli obiettivi di apprendimento.

0.6 L'Esame per il Certificato Advanced Level Test Management

L'esame per il Certificato Advanced Level Test Management si basa su questo Syllabus. Le risposte alle domande d'esame possono richiedere l'uso di materiale basato su più di un paragrafo di questo Syllabus. Tutti i paragrafi del Syllabus sono esaminabili, ad eccezione dell'Introduzione, delle Appendici e dei Riferimenti. Gli standard e i libri sono inclusi come riferimenti (Capitolo 5), ma il loro contenuto non è esaminabile, al di là di quanto sintetizzato in questo Syllabus relativamente a tali standard e libri.

Per ulteriori dettagli, fare riferimento al documento *'Exam Structures and Rules 1.1 Compatible with Syllabus Foundation and Advanced Levels and Specialists Modules'*.

- Il criterio di accesso per sostenere l'esame di certificazione Advanced Level Test Management è il seguente:
 - I candidati devono essere in possesso del Certificato ISTQB® Foundation Level prima di sostenere l'esame di certificazione Advanced Level Test Management. Tuttavia, è fortemente consigliato che il candidato abbia almeno un minimo di background nello sviluppo software o nel testing del software, come ad esempio sei mesi di esperienza come tester o come sviluppatore software.

0.7 Accredитamento

Un Member Board ISTQB® può accreditare Training Provider, il cui materiale didattico segue questo Syllabus. I Training Provider dovrebbero ottenere le linee guida per l'accreditamento dal Member Board o dall'ente che effettua l'accreditamento. Un corso accreditato viene riconosciuto essere conforme a questo Syllabus ed è autorizzato a gestire un esame ISTQB® come parte del corso stesso.

Le linee guida per l'accreditamento di questo Syllabus seguono le *Accreditation Guidelines* pubblicate dal Working Group ISTQB® *Processes Management and Compliance*.

0.8 Gestione degli Standard

Il Syllabus Advanced Level Test Management contiene riferimenti a standard (p.e. ISO, IEC e IEEE). Lo scopo di questi riferimenti è fornire un framework o una fonte di informazioni aggiuntive, se il lettore lo

desidera. Si noti che il Syllabus utilizza questi standard come riferimenti. Il contenuto degli standard non è oggetto di esame. Per ulteriori informazioni sugli standard, fare riferimento al Capitolo 5.

0.9 Livello di Dettaglio

Il livello di dettaglio di questo Syllabus consente di organizzare corsi ed esami consistenti a livello internazionale. Per raggiungere questo obiettivo, il Syllabus consiste di:

- Obiettivi didattici generali che descrivono i propositi del Syllabus Advanced Level Test Management.
- Una lista di termini (parole chiave) che gli studenti devono essere in grado di ricordare.
- Obiettivi di apprendimento per ogni area di conoscenza, che descrivono i risultati cognitivi di apprendimento che devono essere raggiunti.
- Una descrizione dei concetti chiave, che include i riferimenti a fonti riconosciute come accettate dalla letteratura o dagli standard.

Il contenuto del Syllabus non è una descrizione dell'intera area di conoscenza del testing del software, ma riflette il livello di dettaglio che deve essere coperto dai corsi di formazione Advanced Level Test Management. Si focalizza sui concetti e sulle tecniche di test che possono essere applicati a tutti i progetti software.

0.10 Come è Organizzato questo Syllabus

Sono presenti tre capitoli con contenuti esaminabili. Nell'intestazione principale di ogni capitolo è specificato il tempo minimo richiesto ai corsi di formazione accreditati per coprire i contenuti del capitolo; la tempistica non viene fornita per i sottocapitoli interni ad ogni capitolo. Per i corsi di formazione accreditati, il Syllabus richiede un minimo di 22,75 ore di istruzione, distribuite nei tre capitoli come segue:

- Capitolo 1: Gestire le Attività di Test (750 minuti)
 - Il partecipante impara a spiegare le attività di test management (p.e. pianificazione dei test, monitoraggio dei test, controllo dei test e completamento dei test)
 - Il partecipante impara a definire una strategia di test del progetto compresi gli obiettivi del test e la selezione dell'approccio del test appropriato, consistente con la strategia di test dell'organizzazione e con il contesto di progetto
 - Il partecipante impara a gestire progetti in diversi contesti
 - Il partecipante impara ad applicare il testing basato sul rischio per focalizzare il testing sui rischi identificati
 - Il partecipante impara a guidare le attività di test process improvement (miglioramento del processo di test) conducendo una retrospettiva del progetto o dell'iterazione
 - Il partecipante apprende come migliorare il supporto degli strumenti per il testing prendendo in considerazione i rischi, i costi e i benefici del supporto degli strumenti
- Capitolo 2: Gestire il prodotto (390 minuti)
 - Il partecipante apprende come monitorare e controllare il testing per raggiungere gli obiettivi del test utilizzando le metriche di test, e per valutare e fornire report sull'avanzamento del test

- Il partecipante impara a selezionare le corrette tecniche di stima del test per differenti ambiti, seguendo diversi modelli dello sviluppo software
- Il partecipante apprende come definire un workflow del difetto nell'ambito del defect management, per adattarsi al modello dello sviluppo software sequenziale, Agile e ibrido
- Capitolo 3: Gestire il team (225 minuti)
 - Il partecipante apprende come analizzare un determinato contesto di progetto per identificare le competenze richieste al team di test
 - Il partecipante impara a gestire un team secondo l'approccio whole-team
 - Il partecipante apprende come definire un business case per le attività del testing nel progetto

Nota: In questo Syllabus, per ogni Obiettivo di Apprendimento (LO, Learning Objective) esiste un sotto-paragrafo corrispondente con i relativi contenuti (p.e. per LO-1.2.3 esiste un sotto-paragrafo 1.2.3).

0.11 Quali sono le Assunzioni Fondamentali di questo Syllabus?

Questo Syllabus è rivolto a chiunque desideri raggiungere un livello avanzato di competenza nel test management, come test manager, test analyst, test engineer, consulenti del test, coordinatori del test, test leader e project manager. Il Syllabus è allineato al Syllabus ISTQB® Foundation Level V.4, che fornisce le conoscenze di base e la comprensione del testing del software.

Questo Syllabus copre due ruoli principali nel testing: il ruolo di test management e il ruolo di testing. Il ruolo di test management è noto anche come test manager nel contesto del modello di sviluppo sequenziale, dove il test manager è tipicamente un ruolo separato dal project manager o dal responsabile di prodotto. Il ruolo di test management è responsabile del processo di test complessivo, del team di test e del test management. Questo include la definizione della strategia di test, la pianificazione delle attività di test, il monitoraggio e il controllo dell'avanzamento del test, il reporting dei risultati del test, la gestione dei rischi e dei problemi del test. Il ruolo di test management assicura anche che gli obiettivi del test siano allineati con le esigenze del business e degli stakeholder, e che le attività di test siano coordinate con gli altri stakeholder di progetto.

Il ruolo di testing svolge anche attività di valutazione del test, di defect management e di chiusura del test. Il ruolo di testing utilizza varie tecniche del testing per garantire la qualità e l'affidabilità degli artefatti del test e del system under test. Il ruolo di testing utilizza anche strumenti di test e la test automation per supportare il processo di test e migliorare l'efficienza e l'efficacia del test. Le attività e i compiti assegnati a questi ruoli possono variare in base al contesto, come il progetto, il prodotto, le competenze e l'organizzazione. (si veda il Syllabus ISTQB® Foundation Level V.4).

Il termine *membro del team di test* viene utilizzato in questo Syllabus per riferirsi a qualsiasi persona che ricopre un ruolo di test management o di testing che esegue il testing, indipendentemente dal contesto organizzativo e dagli altri ruoli. I team di test sono composti da individui con abilità e competenze differenti. I membri del team possono anche avere diversi livelli di esperienza e di certificazione, come il foundation level, advanced level o expert level. I membri del team di test possono anche avere ruoli e responsabilità differenti in base all'approccio del test e al modello del processo di test utilizzato, come il testing Agile, il testing model-based, il testing basato sul rischio, ecc.

Un punto importante sulla prospettiva che il Syllabus fornisce è che si focalizza sul test management a livello di progetto e non sul test management a livello di organizzazione. Pertanto, questo Syllabus è conforme e contiene informazioni che possono essere utilizzate per le attività di test management a livello

di progetto, ma meno per il test management a livello di organizzazione.

Lo sviluppo software ibrido viene utilizzato in questo Syllabus per indicare un approccio dello sviluppo software che combina elementi di differenti modelli del ciclo di vita del software, come V-model, iterativo, incrementale e Agile. Lo sviluppo software ibrido ha l'obiettivo di sfruttare i punti di forza e mitigare le debolezze di ogni modello, in base al contesto e alle esigenze di progetto. Ad esempio, un approccio allo sviluppo ibrido del software può utilizzare un V-model per le fasi iniziali di pianificazione e analisi dei requisiti, seguito da un modello Agile per le fasi di progettazione, sviluppo e testing. In alternativa, un approccio dello sviluppo software ibrido può utilizzare un modello iterativo per la gestione complessiva del progetto, mentre applica un modello incrementale per ogni iterazione, e un modello Agile per ogni incremento. Lo sviluppo software ibrido richiede un elevato grado di flessibilità, comunicazione e collaborazione tra gli stakeholder, oltre a una chiara comprensione degli obiettivi, dei rischi e dei vincoli di ogni fase e modello.

In base a questo Syllabus e al Glossario, una strategia di test è una descrizione di come il testing verrà eseguito per raggiungere gli obiettivi del test in determinate circostanze. Una strategia di test definisce l'ambito complessivo, l'approccio e le risorse per il testing di un sistema o di un prodotto. È generalmente documentata in un test plan o come parte di altri documenti, in base al contesto del testing. Una strategia di test è influenzata dalla strategia di test dell'organizzazione, che è una strategia di test di alto livello che descrive come il testing viene eseguito in un'organizzazione. Una strategia di test può anche esistere per un singolo livello di test o per un tipo di test, che sono aspetti specifici del testing che si focalizzano su differenti obiettivi, target e criteri. Il termine generico "strategia di test" può essere utilizzato in qualsiasi contesto (progetto, organizzazione, prodotto). Un approccio del test è il modo in cui vengono implementati i compiti del testing, in particolare la selezione e la combinazione dei livelli di test, dei tipi di test e delle tecniche di test per il testing statico e dinamico, oltre ad altre pratiche di test come il testing guidato dagli script, il testing manuale, il testing back-to-back, ecc. L'approccio del test scelto dal ruolo di test management è una decisione chiave nella formulazione di una strategia di test appropriata per un determinato contesto.

1 Gestire le Attività di Test - 750 minuti

Parole Chiave

analisi del rischio, ciclo di vita dello sviluppo software, completamento dei test, controllo dei test, identificazione del rischio, impatto del rischio, livello di rischio, livello di test, metodologia degli obiettivi S.M.A.R.T., mitigazione del rischio, modello di sviluppo incrementale, modello di sviluppo iterativo, modello di sviluppo sequenziale, modello di sviluppo software ibrido, monitoraggio dei test, monitoraggio del rischio, obiettivo del test, pianificazione dei test, probabilità del rischio, retrospettiva, rischio di prodotto, rischio di qualità, risk management, strategia di test, Test Maturity Model integration, test plan, test process improvement, testing basato sull'esperienza, testing basato sul rischio, testing funzionale, testing non-funzionale, tipo di test, TPI NEXT, valutazione del rischio

Parole Chiave Specifiche del Dominio

Goal-Question-Metric (GQM), IDEAL, indicatore, misura, metrica

Obiettivi di Apprendimento per il Capitolo 1:

1.1 Il Processo di Test

- TM-1.1.1 (K2) Riassumere la pianificazione dei test
- TM-1.1.2 (K2) Riassumere il monitoraggio dei test e il controllo dei test
- TM-1.1.3 (K2) Riassumere il completamento dei test

1.2 Il Contesto del Testing

- TM-1.2.1 (K2) Confrontare i motivi per cui differenti stakeholder sono interessati al testing
- TM-1.2.2 (K2) Spiegare perché la conoscenza degli stakeholder è importante nel test management
- TM-1.2.3 (K2) Spiegare il testing in un modello di sviluppo software ibrido
- TM-1.2.4 (K2) Riassumere le attività di test management per i diversi cicli di vita dello sviluppo software
- TM-1.2.5 (K2) Confrontare le attività di test management con i diversi livelli di test
- TM-1.2.6 (K2) Confrontare le attività di test management per i diversi tipi di test
- TM-1.2.7 (K4) Analizzare un progetto e determinare le attività di test management che enfatizzano la pianificazione dei test, il monitoraggio dei test e il controllo dei test

1.3 Testing Basato sul Rischio

- TM-1.3.1 (K2) Spiegare le varie misure adottate nel testing basato sul rischio per poter rispondere ai rischi
- TM-1.3.2 (K2) Fornire esempi di tecniche differenti che un test manager può utilizzare per identificare i rischi relativi alla qualità del prodotto
- TM-1.3.3 (K2) Riassumere i fattori che determinano i livelli di rischio relativi alla qualità del prodotto
- TM-1.3.4 (K4) Selezionare le attività di test appropriate per mitigare i rischi in base al loro livello di rischio in un determinato contesto

- TM-1.3.5 (K2) Distinguere tra esempi heavyweight e lightweight di tecniche del testing basato sul rischio
- TM-1.3.6 (K2) Fornire esempi di metriche di successo e di difficoltà associate al testing basato sul rischio

1.4 La Strategia di Test del Progetto

- TM-1.4.1 (K2) Spiegare le scelte tipiche per un approccio del test
- TM-1.4.2 (K4) Analizzare una strategia di test dell'organizzazione e il contesto di progetto per selezionare l'approccio del test appropriato
- TM-1.4.3 (K3) Utilizzare la metodologia degli obiettivi S.M.A.R.T. per definire obiettivi del test e criteri di uscita misurabili

1.5 Migliorare il Processo di Test

- TM-1.5.1 (K2) Spiegare come utilizzare il modello IDEAL per il test process improvement (miglioramento del processo di test) in un determinato progetto
- TM-1.5.2 (K2) Riassumere l'approccio di miglioramento model-based al test process improvement (miglioramento del processo di test) e comprendere come applicarlo in un contesto di progetto
- TM-1.5.3 (K2) Riassumere l'approccio di miglioramento analytical-based al test process improvement (miglioramento del processo di test) e comprendere come applicarlo in un contesto di progetto
- TM-1.5.4 (K3) Implementare una retrospettiva di progetto o di iterazione per valutare i processi di test e scoprire le aree del testing da migliorare

1.6 Strumenti di Test

- TM-1.6.1 (K2) Riassumere le best practice per l'introduzione dello strumento
- TM-1.6.2 (K2) Spiegare l'impatto dei diversi aspetti tecnici e di business nella scelta di un tipo di strumento
- TM-1.6.3 (K4) Analizzare una determinata situazione per creare un piano per la selezione di uno strumento, che copra i rischi, i costi e i benefici
- TM-1.6.4 (K2) Distinguere le fasi del ciclo di vita dello strumento
- TM-1.6.5 (K2) Fornire esempi per la raccolta e la valutazione di metriche utilizzando gli strumenti

1.1 Il Processo di Test

Introduzione

Il Syllabus ISTQB® Foundation Level V.4 descrive un processo di test che comprende le seguenti attività: pianificazione dei test, monitoraggio e controllo dei test, analisi dei test, progettazione dei test, implementazione dei test, esecuzione dei test e completamento dei test.

Il Syllabus ISTQB® Foundation Level V.4 afferma che queste attività del processo di test sono spesso implementate iterativamente o in parallelo, in base al modello del ciclo di vita dello sviluppo software (Software Development Lifecycle o SDLC) e al contesto di progetto. Generalmente è necessario eseguire un tailoring (adattamento) di queste attività all'interno del contesto del prodotto e del progetto.

Questo Syllabus si focalizza sulle seguenti attività chiave di test management:

- **Pianificazione dei test:** definire gli obiettivi del test, l'approccio del test, l'ambito del test, le risorse del test, la schedulazione del test, i deliverable del test e i partecipanti del test (stakeholder del test).
- **Monitoraggio e controllo dei test:** tracciare l'avanzamento del test, i risultati del test e le deviazioni del test; intraprendere azioni correttive quando necessario; generare report sullo stato e sui risultati del test agli stakeholder rilevanti.
- **Completamento dei test:** finalizzare e archiviare gli artefatti del test, valutare il processo di test e il prodotto del test, identificare le azioni di test process improvement (miglioramento del processo di test) e comunicare la chiusura del test agli stakeholder rilevanti.

Lo standard ISO/IEC/IEEE 29119-2 definisce i processi di test management che coprono queste attività di test management. Questi processi di test management possono essere applicati a diversi livelli del testing, come a livello di progetto, programma o portfolio. Ogni livello del testing può avere un proprio test plan che è allineato al test plan di livello più alto.

1.1.1 Attività di Pianificazione dei Test

Questo paragrafo si focalizza sulle attività di pianificazione dei test per diversi ambiti, come un intero progetto, un livello di test, un tipo di test o una release/iterazione/sprint in Agile. A seconda dell'ambito, la pianificazione dei test può iniziare e terminare in punti differenti del processo di sviluppo. La pianificazione dei test è un'attività che comporta l'identificazione delle attività e delle risorse necessarie per raggiungere gli obiettivi del test che sono stati identificati nella politica di test. La pianificazione dei test dovrebbe iniziare il prima possibile nel processo di sviluppo, preferibilmente prima che i requisiti vengano identificati, e dovrebbe essere tenuta aggiornata man mano che il progetto avanza. La pianificazione dei test è spesso un processo iterativo che richiede una ripianificazione nel corso del progetto, per adeguarsi alle modifiche e ai feedback.

I seguenti compiti fanno parte della pianificazione dei test (similari a quelli riportati nello standard ISO/IEC/IEEE 29119-2):

- **Comprendere il contesto e organizzare la pianificazione dei test**

Comprendere il contesto dell'organizzazione (p.e. la politica di test e la strategia di test dell'organizzazione), l'ambito del test e l'oggetto di test (cioè il prodotto di lavoro da testare) è fondamentale per la pianificazione dei test. (si veda il paragrafo 1.2., *Il Contesto del Testing*). Questo comporta anche tutte le attività necessarie per organizzare lo sviluppo del test plan e per ottenere l'approvazione di queste attività e della schedulazione da parte degli stakeholder (p.e. il product owner,

il project manager o il manager del team di sviluppo).

- **Identificare e analizzare i rischi di prodotto**

L'analisi dei rischi comporta l'identificazione e la valutazione del potenziale impatto e della probabilità dei rischi di prodotto come parte della pianificazione dei test. Per ulteriori dettagli sui rischi di prodotto, si veda il paragrafo 1.3, *Testing basato sul Rischio*.

- **Identificare gli approcci alla gestione dei rischi**

Sulla base dell'analisi dei rischi, vengono selezionati e documentati nel test plan gli approcci appropriati per la gestione dei rischi. Questi possono includere azioni preventive, correttive o di mitigazione per indirizzare i rischi identificati.

- **Definire l'approccio del test, stimare e allocare le risorse del test**

Sulla base della strategia di test dell'organizzazione, degli standard normativi, dei vincoli imposti dal progetto e degli approcci alla gestione dei rischi, viene definito l'approccio del test per l'attuale ambito del testing (si veda il paragrafo 1.4, *La Strategia di Test del Progetto*). Una volta definito l'approccio del test, è importante stimare le risorse del test necessarie, come lo staff del test, gli strumenti di test, gli ambienti di test e i dati di test, e allocare tali risorse alle attività di test.

- **Stabilire il test plan**

Il test plan deve essere accettato da tutti gli stakeholder e, pertanto, i disaccordi tra di loro dovrebbero essere risolti.

1.1.2 Attività di Monitoraggio e Controllo dei Test

Affinché il Test Management possa fornire un controllo dei test efficiente, è necessario definire un framework della schedulazione e del monitoraggio del testing che consenta di tracciare lo stato e l'avanzamento del testing. Questo framework dovrebbe includere le misure e gli obiettivi dettagliati, che sono richiesti per mettere in relazione lo stato dei prodotti di lavoro e delle risorse di test con il test plan e gli obiettivi strategici.

Per progetti piccoli e meno complessi, può essere relativamente semplice mettere in relazione i prodotti di lavoro e le attività di test con il test plan e gli obiettivi strategici, ma in genere devono essere definiti obiettivi più dettagliati per ottenerlo. Questo può includere la definizione di misure e di obiettivi che sono richiesti per soddisfare gli obiettivi del test e la copertura della base di test.

Di particolare importanza è la necessità di correlare lo stato dei prodotti di lavoro e delle attività di test in modo comprensibile e rilevante agli stakeholder di progetto e di business.

Il monitoraggio e il controllo dei test sono attività che vengono svolte in modo continuativo.

Il controllo dei test confronta l'avanzamento effettivo rispetto al test plan, e implementa le azioni correttive necessarie. Guida il testing per soddisfare le strategie di test e gli obiettivi del test (si veda il paragrafo 1.4, *La Strategia di Test del Progetto*) e rivede le attività di pianificazione dei test quando necessario. I dati di controllo richiedono informazioni dettagliate sulla pianificazione dei test per poter reagire in modo appropriato. Questa attività include:

- Implementazione del test plan e delle direttive di controllo
- Gestione delle deviazioni dal testing pianificato
- Gestione dei rischi recentemente identificati oppure modificati

- Valutazione della readiness per iniziare il testing
- Garanzia di ottenere l'approvazione del completamento dei test in base ai criteri di uscita

Il monitoraggio dei test comporta la raccolta e la registrazione dei risultati del test, l'identificazione delle deviazioni dal testing pianificato, l'identificazione e l'analisi di nuovi rischi che richiedono il testing, e il monitoraggio delle modifiche per i rischi identificati.

1.1.3 Attività di Completamento dei Test

Il completamento dei test avviene generalmente in corrispondenza delle milestone di progetto (p.e. un rilascio, la fine di un'iterazione o il completamento di un livello di test). Per qualsiasi difetto non risolto, vengono create change request o elementi del product backlog. Si veda il Syllabus ISTQB® Foundation Level V.4. Una volta soddisfatti i criteri di uscita, i risultati chiave dovrebbero essere catturati, archiviati e forniti agli stakeholder rilevanti. Il completamento dei test richiede i seguenti compiti:

- **Creare e approvare il test completion report**

Questo compito assicura che tutto il testing sia stato eseguito e che tutti gli obiettivi del test siano stati raggiunti. Prevede la raccolta di informazioni rilevanti da diversi testware come i test plan, i risultati del test, i test progress report, i test completion report e i defect report. Le informazioni raccolte vengono valutate e riassunte nel test completion report. Il test completion report viene approvato e comunicato agli stakeholder rilevanti.

- **Archiviare il testware**

Questo compito identifica il testware che può essere utile in futuro, o che si prevede di riutilizzare, che sono in genere i test case. Li rende accessibili e facili da comprendere per un futuro riutilizzo. Inoltre, i risultati del test, i test log, i test report e altro testware dovrebbero essere temporaneamente archiviati nel sistema di configuration management.

- **Consegnare il testware**

Questo compito consegna i prodotti di lavoro essenziali a coloro che ne necessitano. Ad esempio, i difetti noti, differiti o accettati, dovrebbero essere comunicati a coloro che utilizzeranno o supporteranno l'uso del testware.

- **Eseguire tutte le operazioni necessarie per pulire l'ambiente di test e ripristinarlo a uno stato predefinito**

Questo compito assicura che l'ambiente di test sia pronto per il successivo ciclo di testing o progetto. Comporta l'eliminazione di tutti i dati di test, degli strumenti di test, dei driver del test, degli stub del test e dei test script dall'ambiente di test. Comporta anche il ripristino dell'ambiente di test allo stato originale o desiderato.

- **Eseguire/raccogliere/documentare le lessons learned**

Questo compito viene svolto durante le retrospettive, dove vengono discusse e documentate le lezioni importanti che sono state apprese durante il processo di test. Questo può includere risultati per l'intero ciclo di vita dello sviluppo software (SDLC, Software Development Lifecycle). Le lessons learned possono essere utilizzate per migliorare il processo di test, come descritto nel paragrafo 1.5, *Migliorare il Processo di Test*.

1.2 Il Contesto del Testing

Introduzione

Il contesto del testing comprende le uniche condizioni e vincoli che influenzano il processo di test, modellando le decisioni e le strategie per la pianificazione, la progettazione e l'esecuzione dei test. È fondamentale che i test manager comprendano questo contesto per allineare il testing alle esigenze e agli obiettivi specifici del progetto di sviluppo software. Questo contesto può variare in base al tipo di prodotto, all'industria, ai requisiti normativi e, soprattutto, al ciclo di vita dello sviluppo software che viene utilizzato.

I test manager hanno il compito di applicare strategie di test consolidate e di scegliere le tecniche di test piuttosto che svilupparle. Svolgono un ruolo chiave nella formulazione di test plan che sono adattati al contesto di progetto. Comprendendo e considerando questi diversi fattori, i test manager possono garantire che il testing sia pertinente, efficace ed efficiente nel raggiungere gli obiettivi del test.

1.2.1 Stakeholder del Test

Gli stakeholder del test sono individui o gruppi con un interesse diretto o indiretto nella qualità del prodotto. Di seguito è riportata una lista tipica di potenziali stakeholder, che viene modificata per riflettere i loro diversi interessi nel testing:

- **Sviluppatori, Development Lead e Development Manager:** Oltre a implementare il sistema sotto test e ad agire sui risultati del test, questi stakeholder sono coinvolti anche durante lo unit testing e contribuiscono al processo del testing.
- **Tester, Test Lead e Test Manager:** Questi individui preparano il testware, che include sviluppare i test plan e contribuire al processo del testing attraverso delle attività come l'analisi dei requisiti, la progettazione dei test, l'esecuzione dei test, il tracciamento e il reporting dei difetti, la test automation e il reporting dell'avanzamento del test.
- **Project Manager, Product Owner e Utenti di Business:** Questi individui specificano i requisiti, definiscono il livello di qualità richiesto e raccomandano la copertura richiesta in base ai rischi percepiti. Inoltre, eseguono la review dei prodotti di lavoro, partecipano allo User Acceptance Testing (UAT) e prendono decisioni in base ai risultati del test.
- **Team di Operations:** Questi individui sono coinvolti durante l'Operational Acceptance Testing, assicurano la readiness del sistema per il rilascio in produzione e contribuiscono alla definizione dei requisiti non-funzionali.
- **Clienti e Utenti:** I clienti acquistano il prodotto, mentre gli utenti lo utilizzano direttamente. Entrambi sono fondamentali nella definizione dei requisiti e dovrebbero essere coinvolti nello User Acceptance Testing per verificare che il prodotto soddisfi le loro esigenze.

Questa lista non comprende tutti i potenziali stakeholder. I test manager devono condurre un'analisi degli stakeholder come parte della creazione della strategia di test e del test plan, considerando l'ambito del test nell'identificare gli stakeholder specifici per il loro progetto.

1.2.2 Importanza della Conoscenza degli Stakeholder nel Test Management

Nel test management, è cruciale considerare le prospettive e l'influenza dei vari stakeholder. La matrice degli stakeholder, spesso indicata come matrice potere-interesse, guida i test manager nel prioritizzare il

coinvolgimento degli stakeholder e nel gestire le aspettative in modo efficiente. La matrice degli stakeholder è uno strumento strategico per il test management che:

- Utilizza le competenze degli stakeholder, con utenti finali e team tecnici che forniscono feedback e approfondimenti su prestazioni e sicurezza.
- Supporta il risk management evidenziando gli interessi e l'influenza degli stakeholder, e incoraggiando sforzi proattivi di mitigazione.
- Valorizza le diverse prospettive con un feedback di valore.

La matrice degli stakeholder è composta da quattro quadranti:

- **Promoter (Alta Influenza, Alto Interesse):** collaboratori chiave con alta influenza e alto interesse, fondamentali per dare forma alla strategia di test e al test plan.
- **Latenti (Alta Influenza, Basso Interesse):** anche se non hanno un forte interesse nelle attività quotidiane, le loro decisioni sono fondamentali per l'allocazione delle risorse e per la direzione del progetto ad alto livello.
- **Difensori (Bassa Influenza, Alto Interesse):** spesso forniscono un feedback qualitativo e possono essere tenuti impegnati attraverso aggiornamenti regolari e coinvolgimenti appropriati in discussioni specifiche.
- **Apatici (Bassa Influenza, Basso Interesse):** anche se non sono strettamente coinvolti, è importante tenerli aggiornati sulle milestone più importanti e cercare il loro feedback su problemi particolari, per cui possono dare indicazioni uniche e importanti.

Il ruolo del test manager include la compilazione di un elenco dettagliato degli stakeholder e la comprensione della relazione di ognuno di loro con le attività del testing, utilizzando la matrice degli stakeholder per migliorare l'efficacia delle pratiche di test management.

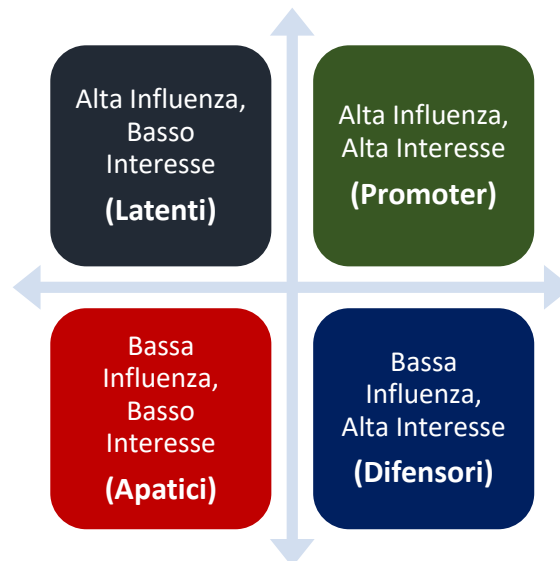


Figura 1. Diversi tipi di stakeholder

1.2.3 Il Test Management in un Modello di Sviluppo Software Ibrido

I modelli di sviluppo software ibridi integrano elementi sia dagli approcci sequenziali tradizionali sia dalle pratiche Agile, per soddisfare esigenze specifiche del progetto oppure transizioni organizzative. Le ragioni tipiche per l'utilizzo di un modello di sviluppo software ibrido sono le seguenti anche se, a seconda dell'organizzazione e del progetto, possono esistere altre ragioni:

- **Ibrido inteso come transizione verso Agile:** la transizione dalle metodologie tradizionali alle metodologie Agile può essere sfidante a causa dei cambiamenti fondamentali nel workflow, nella cultura e nelle dinamiche del team. I modelli ibridi forniscono un approccio bilanciato che facilita questa transizione, combinando la struttura dei metodi tradizionali con la flessibilità delle pratiche Agile.
- **Ibrido inteso come fit-for-purpose:** alcune organizzazioni o progetti possono non essere in grado di passare alle metodologie Agile. I progetti che sono ad alto rischio possono richiedere attività sequenziali per alcuni aspetti e pratiche Agile per altri aspetti. Possono utilizzare un modello ibrido in quanto adatto al loro scopo (fit-for-purpose).

In un contesto ibrido, le attività di test management possono includere:

- Valutare la comprensione e la capacità del team di eseguire senza problemi una transizione dalle metodologie tradizionali alle metodologie Agile
- Identificare i punti di forza e di debolezza nell'adattamento a un approccio ibrido
- Garantire che il team sia in grado di combinare processi strutturati con la flessibilità Agile
- Migliorare la collaborazione tra il team di test e gli stakeholder, per gestire meglio il testing all'interno degli sprint/iterazioni e delle fasi di test tradizionali
- Partecipare a sforzi coordinati, come lo scrum-of-scrums per i tester, per mantenere il focus sul testing e contribuire al contempo agli obiettivi complessivi dello sviluppo
- Tracciare ed eseguire la review dell'effort del test e dei test case all'interno di sprint/iterazioni, per garantire l'allineamento con le pratiche Agile

Ulteriori informazioni possono essere trovate in (Fowler, 2010).

1.2.4 Le Attività di Test Management per i Diversi Modelli del Ciclo di Vita dello Sviluppo Software

Per allineare in modo appropriato il testing al modello del ciclo di vita dello sviluppo software (SDLC), il test manager deve comprendere i diversi modelli del SDLC utilizzati nella propria organizzazione e utilizzare questa conoscenza per allineare in modo appropriato il testing alle attività di sviluppo.

La tabella seguente mostra un confronto tra le diverse attività di test management in base ai diversi modelli del SDLC:

Aspetto	Modello di Sviluppo Sequenziale p.e. V-Model	Modello di Sviluppo Iterativo p.e. SCRUM
Stima	Stima anticipata e dettagliata per ogni livello di test.	Stima iterativa, pianificazione delle sole user story dell'iterazione.
Testware	Include strategia di test, test plan, test case, schedulazione del test e test report.	Si focalizza sui criteri di accettazione e sulla definition of done; documentazione minima.
Ruoli	Il test manager supervisiona le decisioni e la gestione del team.	I ruoli sono integrati; un facilitatore o un coach sostituisce il tradizionale test manager.
Strumenti	Strumenti di test management prevalentemente adatti per il testing basato su fasi.	Gli strumenti per la CI/CD (Continuous Integration/Continuous Deployment) e la test automation sono centrali e supportano il continuous testing.
Approccio del Testing	Schedulato in anticipo, corrispondente alle fasi del progetto.	Incorporato nelle iterazioni, con un'attenzione particolare all'adattabilità e al feedback.
Test automation	Implementato in modo strategico, può avvenire in diverse fasi.	Integrato già nella fase iniziale del progetto (inception), con enfasi sul regression testing automatizzato nella CI/CD.
Monitoraggio e reporting	Reporting basato su milestone, con dashboard automatizzate opzionali.	Reporting continuo con dashboard real-time e aggiornamenti quotidiani sullo stato.
Metriche	Si focalizza sulle metriche di test tradizionali e sul defect management. (p.e. esecuzione dei test, percentuale di difetti).	Include metriche Agile per il tracciamento delle iterazioni in aggiunta alle metriche tradizionali. (p.e. velocità del team, burndown chart).

Tabella 1: Attività di Test Management per Diversi Modelli del SDLC

1.2.5 Attività di Test Management nei Diversi Livelli di Test

Testing di componente:

- Definire l'ambito, gli obiettivi e i criteri di completamento del testing di componente (unit testing).
- Coinvolgere i tester in attività che vanno oltre i ruoli tradizionali del testing, come le code review, dove le loro competenze analitiche aggiungono valore.
- Coordinarsi con il team di sviluppo per la risoluzione dei problemi e il contributo allo unit test.

Testing di integrazione dei componenti:

- Determinare le sequenze di integrazione e le combinazioni dei test in collaborazione con il team di sviluppo, tenendo conto del modello del SDLC, degli strumenti e dei processi.
- Supervisionare l'avanzamento per garantire che sia allineato alle strategie del testing di sistema e del testing di accettazione.

- Gestire questa fase in collaborazione con gli sviluppatori, considerando anche il testing di integrazione dei componenti (unit).

Testing di integrazione dei sistemi:

- Garantire che l'ambito e gli obiettivi del testing di integrazione dei sistemi siano chiari e in sintonia con la valutazione del rischio e con gli obiettivi di qualità.
- Mantenere la supervisione dell'avanzamento, dei risultati e della gestione dei problemi durante il testing di integrazione dei sistemi.

Testing di sistema:

- Adattare la pianificazione al modello del SDLC, con un'attenta allocazione delle risorse, un'appropriata selezione degli strumenti e una corretta schedulazione.
 - Per i progetti Agile, integrare il testing di sistema con il testing iterativo delle user story, evitando fasi di test distinte e garantendo che il testing sia continuo e integrato. Nei modelli sequenziali, invece, il testing può seguire fasi pianificate.

Testing di accettazione:

- Collaborare con gli stakeholder per eseguire la review e confermare che siano soddisfatti i criteri di accettazione, e pianificare le attività del testing, compresa la gestione del testing dell'utente durante gli UAT.
- Coordinare la logistica del testing di accettazione, facilitando i test presso la sede del cliente, per garantire che il prodotto soddisfi le esigenze di business e gli standard di qualità fuori dall'ambiente di sviluppo.
- Facilitare la risoluzione di eventuali problemi durante gli UAT e guidare gli stakeholder attraverso il processo di accettazione del prodotto, una volta che i criteri di accettazione sono stati soddisfatti.

1.2.6 Attività di Test Management per Diversi Tipi di Test

Un test management efficace richiede un approccio integrato che tenga conto delle esigenze specifiche del testing funzionale, non-funzionale, black-box e white-box. Per i manager che eseguono il testing funzionale, il focus è garantire che tutte le funzionalità siano testate a fondo e soddisfino i requisiti definiti. Il management del testing non-funzionale si focalizza sulla verifica di attributi del sistema, come le prestazioni e la sicurezza. Il management del testing black-box si focalizza a garantire che i test siano focalizzati sull'utente e che siano coperte tutte le possibili interazioni esterne. Il management del testing white-box enfatizza la comprensione della struttura del codice e la garanzia che i test coprano a fondo la logica interna.

Management del Testing Funzionale:

- Pianificazione Strategica e Tracciamento dell'Avanzamento: Creare una strategia di test dettagliata che sia allineata ai requisiti funzionali e agli obiettivi del progetto, e monitorarne l'avanzamento.
- Coordinamento delle Risorse: Allocare in modo efficiente le risorse umane e tecniche per coprire tutti gli aspetti funzionali del sistema.

Management del Testing Non-Funzionale:

- **Benchmarking delle Prestazioni:** Stabilire i benchmark delle prestazioni e gestire le attività del testing che valutano il sistema rispetto a questi criteri.
- **Verifica della Conformità:** Supervisionare i test che garantiscono la conformità del sistema rispetto a standard non-funzionali come la sicurezza, l'usabilità e l'affidabilità.

Management del Testing Black-Box:

- **Analisi della Copertura dei Test:** Assicurare che i test black-box coprano tutti gli scenari utente e i requisiti di business.
- **Integrazione del Feedback:** Gestire il processo di raccolta del feedback degli stakeholder per raffinare gli approcci del testing black-box e la correzione dei difetti.

Management del Testing White-Box:

- **Ottimizzazione della Copertura del Codice:** Supervisionare l'utilizzo di strumenti di copertura del codice per identificare le lacune nel testing white-box, e gestire le risorse per indirizzare queste aree.
- **Integrazione degli Approfondimenti Tecnici:** Gestire l'integrazione degli approfondimenti tecnici nel processo di pianificazione dei test, garantendo che i test siano progettati con una comprensione del funzionamento interno dell'applicazione.

1.2.7 Attività di Test Management per Pianificare, Monitorare e Controllare

Un test management efficace è la pietra angolare dell'effort di un testing di successo e comprende un'ampia gamma di attività che richiedono un'attenta pianificazione, un monitoraggio vigile e un controllo strategico. I test manager svolgono un ruolo fondamentale nel garantire che il processo di test sia non solo efficace ed efficiente, ma anche adattato rispetto alle esigenze specifiche del progetto in questione.

Pianificazione dei Test:

- **Definizione Completa dell'Ambito:** Un test plan deve essere realizzato in modo preciso, incorporando una definizione approfondita dell'ambito. Questo include l'identificazione di tutti i requisiti funzionali e non-funzionali per garantire una copertura completa del test. Inoltre, occorre considerare le implicazioni di entrambe le metodologie del testing black-box e white-box, assicurando che i test case sviluppati siano in grado di validare il sistema sotto test da tutti i punti di vista.
- **Piano di Valutazione e Mitigazione dei Rischi:** Il test plan è integrato da un solido framework di risk management. I test manager devono intraprendere un'analisi dei rischi dettagliata, individuando le potenziali vulnerabilità e le sfide che potrebbero avere un impatto sia sul flusso di lavoro del progetto sia sul prodotto finale. Lo sviluppo di strategie di mitigazione è cruciale, e comporta una pianificazione preventiva per aggirare o minimizzare efficacemente questi rischi.
- **Strategia di Allocazione delle Risorse:** La pianificazione delle risorse è un altro elemento critico. Questo si estende oltre la semplice allocazione, definendo la struttura del team, delineando i ruoli e stabilendo i protocolli di comunicazione. Negli ambienti in cui i team sono distribuiti, come nei modelli onsite/offsite, questo aspetto diventa particolarmente importante per mantenere la sincronia e garantire una collaborazione senza interruzioni.

Monitoraggio dei Test:

- **Supervisione dell'Esecuzione:** Il monitoraggio svolge un ruolo centrale nel processo di test management. Comporta una review continua dell'esecuzione dei test rispetto al piano stabilito, il tracciamento dello stato di esecuzione dei test case e la gestione di eventuali difetti. L'adeguamento delle priorità di test in base alle valutazioni del rischio e agli sviluppi real-time garantisce che il testing rimanga focalizzato e allineato alle aree più critiche.
- **Ottimizzazione degli Strumenti e degli Ambienti:** La selezione e l'utilizzo appropriato degli strumenti e degli ambienti di test sono fondamentali per supportare la strategia del testing. Il monitoraggio continuo assicura che siano efficacemente integrati nella pipeline della CI/CD, facilitando il testing continuo e i cicli di feedback immediati, che sono fondamentali per il processo di sviluppo Agile.
- **Collaborazione con lo Sviluppo:** Il mantenimento di uno stretto rapporto di lavoro con il team di sviluppo è essenziale per l'efficacia dei risultati dei test. Questa collaborazione dovrebbe supportare un approccio completo al testing, sfruttando le intuizioni provenienti dalle prospettive white-box e black-box per indirizzare preventivamente i potenziali problemi.

Controllo dei Test:

- **Management Adattivo del Processo:** Il controllo dei test consiste nell'adeguare dinamicamente il processo di test in risposta a nuove intuizioni, sfide e dinamiche di progetto in evoluzione. Richiede che un test manager sia reattivo e flessibile, in grado di implementare le modifiche all'approccio del testing in modo che riflettano lo stato attuale del progetto.
- **Management del Quality Gate:** È fondamentale un approccio strutturato al management del quality gate. Questo include definire quello che costituisce un quality gate all'interno del ciclo di vita del testing e prendere decisioni informate sull'avanzamento della fase del testing, che è strumentale al mantenere l'integrità del prodotto.

Focalizzandosi su queste attività specifiche nell'ambito della pianificazione, del monitoraggio e del controllo dei test, i test manager possono garantire che il processo di test sia ben definito, adattabile alle modifiche del progetto e che il prodotto sia conforme sia ai requisiti di progetto sia alle aspettative degli stakeholder.

1.3 Testing Basato sul Rischio

Introduzione

Il testing basato sul rischio comporta l'identificazione, la valutazione, il monitoraggio e la mitigazione dei rischi per guidare il testing. Questi rischi sono identificati da diversi stakeholder e possono essere utilizzati per selezionare e prioritizzare i test. Più alto è il livello di rischio, prima dovrebbe iniziare il testing, e più intenso e prolungato dovrebbe essere l'effort del test.

1.3.1 Il Testing come Attività di Mitigazione del Rischio

Un rischio di prodotto è una situazione potenziale in cui possono esistere problemi di qualità in un prodotto. Quando i test rivelano dei difetti, il testing ha contribuito a mitigare il rischio di prodotto, fornendo la consapevolezza dei difetti e le opportunità di affrontarli prima del rilascio. Quando i test non rilevano difetti, il testing indica che il livello del rischio di prodotto è inferiore a quello previsto.

Tra le altre cose, il test manager ha la responsabilità di fornire una valutazione corretta e affidabile della qualità del prodotto. Questo richiede un coinvolgimento attivo nel risk management di progetto, con un focus sui rischi di progetto relativi alla quality assurance (p.e. requisiti ambigui che portano a problemi importanti nella validazione svolta in fase avanzata, ambienti di test insufficienti che ostacolano l'esecuzione dei test).

Il testing basato sul rischio focalizza il testing sui rischi di qualità. Segue il processo generico di risk management, che consiste nelle seguenti attività principali:

- Analisi del rischio, che consiste nell'identificazione del rischio e nella valutazione del rischio
- Controllo del rischio, che consiste nel monitoraggio del rischio e nella mitigazione del rischio

Queste attività principali sono logicamente organizzate in ordine sequenziale, ma possono sovrapporsi.

Per focalizzare il testing sui rischi di qualità, è necessario che i rischi di qualità vengano identificati e valutati. Per essere più efficace, l'analisi del rischio dovrebbe includere diversi stakeholder. Essendo uno stakeholder principale dell'analisi dei rischi di qualità, il test manager dovrebbe comprendere e monitorare queste attività, ed essere in grado di moderarle.

Il monitoraggio dei test dovrebbe includere il monitoraggio dei rischi. Oltre a monitorare l'evoluzione dei rischi di qualità noti, dovrebbe includere l'analisi di eventuali nuovi rischi di qualità e l'adeguamento del risk register.

Il test manager è una delle tante persone che guidano la mitigazione dei rischi di qualità. La mitigazione del rischio è distribuita su diverse attività di test. Ad esempio, i risultati dell'analisi dei rischi di qualità vengono utilizzati nella pianificazione dei test, per focalizzare il testing sulle aree corrette utilizzando le tecniche corrette. Nell'analisi dei test, i livelli di rischio guidano la selezione delle condizioni di test da coprire. Nell'esecuzione dei test, la prioritizzazione basata sui rischi governa la sequenza di esecuzione dei test.

1.3.2 Identificazione dei Rischi di Qualità

Il compito del test manager è di raccogliere i rischi dagli stakeholder. Gli stakeholder possono identificare i rischi di qualità attraverso una o più delle seguenti tecniche:

- Interviste agli esperti

- Valutazioni indipendenti
- Retrospective
- Workshop sul rischio
- Brainstorming
- Checklist
- Riferimenti a esperienze passate

Coinvolgendo il più ampio campione possibile di stakeholder, il processo di identificazione del rischio spesso identifica la maggior parte dei rischi di prodotto significativi. L'identificazione degli stakeholder da coinvolgere in questa fase è molto importante. È essenziale assicurarsi che la lista degli stakeholder partecipanti sia completo e concordato con il project manager. L'identificazione dei rischi che non tiene conto degli stakeholder chiave può essere molto problematica. Occorre

garantire che tutti gli stakeholder rilevanti abbiano la possibilità di partecipare. Se non potessero partecipare, dovrebbero almeno avere la possibilità di delegare il compito. Quando gli stakeholder chiave non possono essere rappresentati, è possibile utilizzare un kick-off meeting per determinare se sono assenti.

Nel testing basato sul rischio, è importante capire che il rischio non è distribuito in modo uniforme all'interno degli oggetti di test. Ad esempio, i componenti rivolti al cliente (client-facing) di un'applicazione self-service possono presentare rischi di usabilità molto differenti da quelli dei componenti amministrativi. L'identificazione dei rischi individuali dei vari oggetti di test è un compito importante nella pianificazione dei test.

L'identificazione del rischio spesso produce altri sotto-prodotti (cioè l'identificazione di problemi che non sono rischi di prodotto). Esempi includono domande o problemi generali sul prodotto o sul progetto, oppure problemi nei documenti di riferimento come i requisiti e le specifiche di progettazione. Anche i rischi di progetto sono spesso identificati come sotto-prodotto dell'identificazione dei rischi di qualità, ma non sono il focus del testing basato sul rischio. Il test manager può spesso svolgere un ruolo importante nell'evidenziare questi sotto-prodotti e nel far capire che la qualità è una preoccupazione di tutti. Requisiti carenti o mancanti sono spesso indice di un problema più fondamentale nella pianificazione e nella preparazione, poiché la quality assurance è coinvolta in tutto il SDLC.

1.3.3 Valutazione del Rischio di Qualità

Una volta identificati, i rischi possono essere valutati. La valutazione del rischio di qualità include la categorizzazione dei rischi in base al tipo (rischio di prodotto o rischio di progetto) e alle caratteristiche di qualità impattate.

Determinare il livello di rischio comporta generalmente la valutazione, per ogni elemento di rischio, della probabilità di accadimento del rischio e dell'impatto del rischio quando accade. I fattori che influenzano la probabilità del rischio per i rischi di qualità includono:

- Complessità della tecnologia, degli strumenti o dell'architettura del sistema
- Maturità dell'organizzazione
- Problemi del personale rispetto a competenze, disponibilità, motivazione o autonomia nel lavoro, oltre al conoscenza del SDLC in uso

- Conflitti all'interno del team
- Problemi contrattuali con i fornitori
- Team geograficamente distribuiti
- Debole leadership manageriale o tecnica
- Pressione del management su tempi, risorse e budget
- Mancanza di attività di quality assurance nelle fasi iniziali
- Alta frequenza di modifiche della base di test, del prodotto o del personale

I fattori che influenzano l'impatto del rischio includono:

- Frequenza di utilizzo della funzionalità interessata
- Criticità della funzionalità interessata
- Criticità dell'obiettivo di business interessato
- Danni alla reputazione
- Perdita di redditività
- Potenziali perdite o responsabilità economiche, ecologiche o sociali
- Sanzioni legali, civili o penali
- Problemi di interfaccia e integrazione
- Mancanza di soluzioni alternative ragionevoli
- Esigenze di sicurezza fisica (safety)

Il test manager combina la probabilità del rischio e l'impatto del rischio per determinare il livello di rischio.

Se l'analisi del rischio si basa su dati di rischio completi e statisticamente validi, è appropriata una valutazione quantitativa. Ad esempio, la probabilità del rischio può essere espressa in percentuale e l'impatto del rischio come quantità. In questo caso, il livello di rischio può essere calcolato come il prodotto di questi due fattori. In genere, tuttavia, la probabilità e l'impatto del rischio possono essere valutati solo qualitativamente su scale ordinali, come: molto alto, alto, medio, basso o molto basso. I valori di probabilità e impatto del rischio vengono poi combinati attraverso una matrice del rischio per creare un livello di rischio aggregato. Questo livello di rischio aggregato dovrebbe essere interpretato come una valutazione qualitativa e relativa su scala ordinale.

A meno che l'analisi del rischio non sia basata su dati di rischio completi e statisticamente validi, l'analisi del rischio sarà qualitativa, basata sulle percezioni soggettive degli stakeholder della probabilità del rischio e dell'impatto del rischio.

1.3.4 Mitigazione del Rischio di Qualità attraverso un Testing Appropriato

Nello sviluppo del software, il testing è la più importante attività di mitigazione del rischio di qualità e consente di ridurre la probabilità di failure. Altre possibili misure di mitigazione del rischio includono un contingency plan (p.e. fornendo workaround o soluzioni alternative), il trasferimento del rischio a una terza parte (p.e. il fornitore di un componente) o l'accettazione del rischio.

Nella **pianificazione dei test**, il tempo e l'effort associati allo sviluppo e all'esecuzione di un test dovrebbero essere proporzionali al livello di rischio: il testing per i livelli di rischio più alti dovrebbe iniziare prima e utilizzare tecniche di test più rigorose, mentre il testing per i livelli di rischio più bassi può iniziare più tardi e dovrebbe utilizzare tecniche di test meno rigorose. Per mitigare al meglio il rischio complessivo attraverso il testing, il test manager dovrebbe analizzare i seguenti fattori contestuali e selezionare un approccio del test appropriato:

- **Gli elementi di test:** Differenti elementi di test all'interno di un oggetto di test possono avere differenti livelli dello stesso tipo di rischio, quindi un oggetto di test non richiede di essere testato con un rigore uniforme.
- **Le caratteristiche di qualità:** I rischi che influenzano specifiche caratteristiche di qualità dovrebbero essere mitigati da tipi di test associati che richiedono effort del test, ambienti di test e competenze di test specifici.
- **I livelli di test e i tipi di test:** Alcuni rischi possono essere testati solo dinamicamente in particolari livelli di test, altri con il testing statico (p.e. analisi statica e code review per la manutenibilità) o con una combinazione di entrambi (p.e. review dell'architettura e testing dinamico del sistema integrato per le vulnerabilità della sicurezza). Testare ogni elemento di test il prima possibile mitiga il rischio di rilevare difetti critici in uno stato avanzato del ciclo di vita, che causerebbero ritardi e costi di failure interno più alti.
- **Il SDLC:** Le attività di test hanno criteri di ingresso specifici. I diversi SDLC li soddisfano in tempi differenti.
- **Il team di test:** Le persone più qualificate dovrebbero testare gli elementi di test con i livelli di rischio più alti.
- **I requisiti normativi:** Alcuni standard safety (sicurezza fisica) (p.e. lo standard IEC 61508) prescrivono le tecniche di test e la copertura richiesta in base al safety integrity level. Il test manager deve garantire il rispetto di questi standard.

Inoltre, il livello di rischio dovrebbe influenzare le decisioni relative al quality control, come l'uso di review dei prodotti di lavoro quali i test case, il livello di indipendenza del testing dallo sviluppo, e l'estensione del regression testing eseguito.

Durante il **monitoraggio dei test e il controllo dei test**, il testing basato sul rischio consente di eseguire in qualsiasi momento il reporting sull'avanzamento del test in termini di livello di rischio residuo. Questo supporta il team di sviluppo e gli stakeholder nel monitoraggio e nel controllo dello sviluppo del software, incluso prendere decisioni sul rilascio in base al livello di rischio residuo. Questo richiede il reporting dei risultati del test in termini di rischio, in modo comprensibile per gli stakeholder.

Durante l'**implementazione dei test**, la prioritizzazione dei test si basa sui livelli di rischio. Durante l'esecuzione dei test, questo garantisce la copertura anticipata delle aree più critiche e la mitigazione dei rischi di livello più alto.

- In alcuni casi, i test vengono prioritizzati per l'esecuzione in ordine strettamente decrescente rispetto ai livelli di rischio che coprono, a partire dal livello più alto. Questo approccio è chiamato depth-first ed è appropriato quando è importante mitigare i rischi di livello più alto il prima possibile.
- In alternativa, ad almeno un test per ogni rischio viene assegnata la massima priorità. Tutti gli altri test vengono prioritizzati in base al loro livello di rischio. Questo approccio è chiamato breadth-first ed è appropriato quando gli stakeholder vogliono avere una visione complessiva della qualità del prodotto il prima possibile. Nella pratica, il testing inizia spesso con l'approccio depth-first, ma quando il tempo

diventa più limitato, si passa all'approccio breadth-first, testando tutti gli elementi di rischio rimanenti almeno una volta.

Sia che il testing basato sul rischio segua l'approccio depth-first, sia che segua l'approccio breadth-first o che sia combinato, il tempo allocato per il testing può essere consumato senza che tutti i test pianificati vengano eseguiti. A questo punto, il testing basato sul rischio facilita la disponibilità di una raccomandazione giustificata verso il management, per decidere se estendere il testing o accettare il rischio rimanente.

1.3.5 Tecniche per il Testing Basato sul Rischio

Esistono diverse tecniche specifiche con vari gradi di formalità per implementare il testing basato sul rischio. L'adeguatezza di una tecnica dipende da considerazioni relative al progetto, processo e prodotto. Esistono due tipi fondamentali di tecniche: heavyweight o lightweight. Nei sistemi safety-critical, vengono utilizzate molto spesso le tecniche heavyweight. Nelle applicazioni non-safety-critical, vengono di solito utilizzate le tecniche lightweight.

Le tecniche heavyweight sono formali, e utilizzano procedure definite e una documentazione dettagliata. Coinvolgono ampi gruppi di stakeholder. La valutazione del rischio utilizza fattori dettagliati di probabilità del rischio e impatto del rischio, e formule matematiche per calcolare la probabilità del rischio e l'impatto del rischio a partire da tali fattori. Esempi di tecniche heavyweight sono:

- **Hazard Analysis:** Estende il processo analitico a monte, cercando di identificare i pericoli che sono alla base di ogni rischio
- **Cost of Exposure:** Determina, per ogni elemento di rischio di qualità, la probabilità di accadimento di un failure, il costo di una perdita (finanziaria) associata al failure e il costo del testing per identificare tale failure
- **Failure Mode and Effect Analysis (FMEA) e le sue varianti:** Identifica i rischi di qualità, le loro potenziali cause e i loro probabili effetti, e quindi assegna severità, priorità e tassi di rilevamento
- **Fault tree analysis:** Mette in relazione i potenziali failure con i difetti che possono causare il failure, quindi con gli errori che possono causare questi difetti, continuando fino all'identificazione delle diverse root cause

Le tecniche lightweight, invece, sono meno approfondite e richiedono un effort minore da parte del team di test e degli stakeholder. Come le tecniche heavyweight, anche queste tecniche si basano sul coinvolgimento degli stakeholder, utilizzando i risultati dell'analisi del rischio come base per la pianificazione dei test e per le condizioni di test. Tuttavia, il gruppo di stakeholder può non essere così ampio e i fattori di rischio hanno un impatto del rischio e una probabilità del rischio ridotti su una scala ordinale. Alcune di queste tecniche, come Systematic Software Testing (SST) (Craig & Jaskiel, 2002), possono essere utilizzate solo quando vengono fornite le specifiche dei requisiti. Altre tecniche, come Pragmatic Risk Analysis and Management (PRAM) (Black, 2009) e Product Risk Management (PRISMA) (van Veenendaal, 2012), utilizzano i requisiti e/o altre specifiche come input per l'analisi del rischio, ma possono funzionare basandosi completamente sull'input degli stakeholder.

1.3.6 Metriche di Successo e Difficoltà Associate al Testing Basato sul Rischio

In una retrospettiva, il team di test dovrebbe misurare l'estensione dei benefici realizzati dal testing basato sul rischio. In molti casi, questo comporta la risposta ad alcune o a tutte le seguenti domande, attraverso l'uso di metriche e consultazioni:

- Gli stakeholder rilevanti sono stati coinvolti o rappresentati nell'analisi del rischio?
- Il coinvolgimento degli stakeholder nell'analisi del rischio è stato appropriato?
- Se ci sono stati incidenti critici in produzione che indicano che sono sfuggiti difetti critici, questi sono stati risolti?
- La maggior parte dei difetti di priorità alta è stata rilevata nelle prime fasi di esecuzione dei test?
- Il team di test è stato in grado di spiegare i risultati del test agli stakeholder in termini di rischio?
- I test che non sono stati eseguiti avevano un livello di rischio associato inferiore rispetto a quelli eseguiti?

Nella maggior parte dei casi, un testing basato sul rischio eseguito con successo si traduce in una risposta affermativa a tutte queste domande. Nel lungo termine, si dovrebbero fissare obiettivi di process improvement per le metriche di successo, oltre a cercare di migliorare l'efficienza del processo di analisi dei rischi di qualità.

Gestire il rischio incontra spesso difficoltà inaspettate a causa di complessità che vengono spesso trascurate.

- **Difficoltà nella valutazione del livello di rischio:** Può essere molto difficile stimare l'impatto del rischio e la probabilità del rischio. Soluzione: utilizzare i dati storici e chiedere agli stakeholder chiave del progetto la loro valutazione.
- **Inizi entusiasmanti:** Spesso viene trascurato di implementare e mantenere un approccio adeguato del test basato sul rischio, di fronte a un'elevata pressione di successo nel breve termine. Soluzione: monitorare in modo regolare e fare reporting dei rischi agli stakeholder.
- **Déjà vu:** Per ogni progetto viene gestito lo stesso insieme di rischi, e questo porta all'autocompiacimento nei confronti del rischio. Soluzione: coinvolgere le persone giuste nell'identificazione dei rischi e mitigare solo quei rischi ritenuti importanti.
- **I rischi chiave sono mancanti:** La root cause di questo problema è generalmente dovuta al coinvolgimento nel processo di persone inesperte o inappropriate. Soluzione: coinvolgere le persone appropriate e formarle.
- **Abbandono degli stakeholder:** Gli stakeholder possono cambiare nel corso del tempo e possono emergere anche nuovi rischi, quindi l'analisi del rischio è un'attività continua e iterativa e non dovrebbe essere eseguita soltanto all'inizio.

1.4 La Strategia di Test del Progetto

Introduzione

In tutto il presente Syllabus, la strategia di test dell'organizzazione viene considerata come un dato di fatto. Lo sviluppo e la manutenzione di una strategia di test dell'organizzazione sono considerati nel contesto dello standard ISO/IEC/IEEE 29119-3 (dove si fa riferimento a una "pratica del test dell'organizzazione") e dei Syllabi ISTQB® Expert Level - Test Management e ISTQB® Certified Tester Agile Test Leadership at Scale.

Se una strategia di test dell'organizzazione non esiste o non copre gli aspetti richiesti, il test management deve cercare di chiarire i dettagli mancanti con gli stakeholder rilevanti.

Nel contesto di questo paragrafo, la definizione di una strategia di test del progetto è un esempio di qualsiasi tipo di strategia di test dettagliata per un progetto, una release, un prodotto o qualsiasi altro tipo di sviluppo di sistema o iniziativa di acquisizione. Una strategia di test del progetto (definita "strategia di test" nello standard ISO/IEC/IEEE 29119-3) descrive l'approccio al testing in un contesto specifico, in modo da poter raggiungere gli obiettivi dell'organizzazione, in particolare quelli relativi alla qualità del prodotto e alle attività di test. Una strategia di test può esistere anche per un singolo livello di test o tipo di test.

La strategia di test del progetto è il risultato principale della pianificazione dei test di un progetto ed è generalmente documentata in un test plan o come parte di altri documenti. La documentazione della strategia di test è raccomandata, ma non obbligatoria, che sia prodotta nella forma di un test plan formale. La necessità di documentazione dipende dal contesto del testing (si veda il paragrafo 1.2, *Il Contesto del Testing*). Quando un progetto segue un modello di sviluppo sequenziale, la strategia di test del progetto è generalmente documentata, preferibilmente all'interno del test plan (si veda ISO/IEC/IEEE 29119-3). La documentazione è spesso richiesta anche da contratti, accordi, enti normativi o leggi.

1.4.1 Scegliere un Approccio del Test

La strategia di test del progetto guida tutte le attività del testing all'interno di un progetto e dettaglia gli obiettivi, le risorse, le schedulazioni e le responsabilità. Questa strategia deve essere adattata soltanto ai requisiti del progetto. Le decisioni chiave includono la selezione dei livelli di test, dei tipi di test e delle tecniche di test per il testing statico e dinamico e per le altre pratiche di test (p.e. il testing guidato dagli script, il testing manuale, il testing back-to-back).

In teoria, tutti i tipi di test possono essere eseguiti a qualsiasi livello di test, e qualsiasi tecnica di test può essere applicata a ogni tipo di test a qualsiasi livello di test. In pratica, la selezione e la combinazione appropriate di queste scelte hanno un impatto significativo sull'efficacia e sull'efficienza del testing. Ad esempio, la manutenibilità del codice può essere spesso valutata in modo più efficace ed efficiente utilizzando l'analisi statica del codice o la code review. D'altra parte, l'efficienza delle prestazioni può essere valutata meglio attraverso test di sistema guidati dagli script a causa dell'interazione dei componenti interni, oppure l'utilità delle funzionalità può essere validata meglio con gli utenti attraverso test di accettazione manuali sviluppati in modo collaborativo. La scelta dell'approccio migliore per una strategia di test può essere un processo complesso che può essere influenzato dalla strategia di test dell'organizzazione, dal contesto di progetto e da altri aspetti.

La selezione e la combinazione dei livelli di test, dei tipi di test e delle tecniche di test sono quindi fondamentali per un'efficace strategia del testing di progetto, in quanto influenza in modo significativo l'efficienza e l'efficacia del testing.

1.4.2 Analizzare la Strategia di Test dell'Organizzazione, il Contesto di Progetto e Altri Aspetti

La strategia di test dell'organizzazione, il contesto di progetto e altri fattori o vincoli relativi al testing devono essere pienamente compresi, per consentire lo sviluppo di una strategia di test del progetto.

Per scegliere l'approccio del test appropriato, in genere devono essere analizzati i seguenti fattori:

- **Dominio:** Il dominio per il quale il prodotto sarà creato o modificato. Eventuali normative, standard e pratiche specifiche del dominio possono modificare il rigore del testing, la documentazione richiesta e il relativo livello di dettaglio. Ad esempio, nel settore farmaceutico e medico, l'approccio del test spesso enfatizza uno user acceptance testing intensivo focalizzato sui rischi per la salute del paziente, utilizzando test case basati sui requisiti funzionali dell'utente, mentre uno user acceptance testing per le applicazioni assicurative web-based potrebbero focalizzarsi sull'usabilità e sull'aumento della probabilità di nuovi contratti assicurativi attraverso il testing A/B.
- **Obiettivi dell'organizzazione e caratteristiche di qualità generali:** Gli obiettivi organizzativi possono includere la necessità di dimostrare il valore del testing e di aumentare il grado di test automation o le caratteristiche di qualità del processo di test, come il livello di maturità del testing o l'efficienza del rilevamento dei difetti. Questi possono determinare i livelli di test e i tipi di test a cui aderire.
- **Gli obiettivi di progetto e il tipo di progetto:** Gli obiettivi di progetto (p.e. relativi al budget, al tempo e alla qualità) e il tipo di progetto (p.e. lo sviluppo di un prodotto specifico per il cliente oppure orientato al mercato) contengono tipicamente vincoli e rischi, nonché opportunità che influiscono sul testing. Ad esempio, vincoli stretti di tempo e di budget possono richiedere l'utilizzo rigoroso del testing basato sul rischio per assegnare la priorità ai test case per l'esecuzione dei test, mentre lo sviluppo di un prodotto specifico per un cliente può richiedere test che coprano criteri di accettazione contrattuali predefiniti.
- **Risorse del test:** Devono essere considerati tutti i vincoli relativi alla disponibilità delle risorse del test, compresi gli strumenti di test, l'infrastruttura del test, la tecnologia e l'ambiente di sviluppo utilizzati nel progetto, nonché lo staff del testing disponibile e le loro competenze. (si veda il paragrafo 3.1, *Il Team di Test*). Ad esempio, il testing basato sull'esperienza richiede tester con una buona conoscenza di dominio; le applicazioni mobile solitamente devono essere testate su un numero generalmente limitato di dispositivi diversi; l'utilizzo di strumenti del testing può essere limitato dal numero di licenze disponibili.
- **Il modello del ciclo di vita dello sviluppo software utilizzato per il progetto:** Per determinare livelli di test, effort del test, criteri di ingresso e criteri di uscita appropriati, si veda il Syllabus ISTQB® Foundation Level V.4, paragrafi 2.2 e 5.1. Un ciclo di vita del software che applica il continuous integration richiede testing automatizzato in maggiore quantità rispetto allo sviluppo con un unico rilascio che utilizza un modello waterfall (a cascata), e quindi possono essere utilizzati diversi tipi di test e tecniche di test.
- **Interfacce con altri sistemi:** In un sistema di sistemi, è essenziale allineare il testing con altri team o progetti e selezionare livelli di test appropriati, soprattutto per il testing di integrazione dei sistemi. Ad esempio, il testing basato sul rischio aiuta a prioritizzare e a scalare il testing di integrazione dei sistemi.
- **Disponibilità dei dati di test:** Devono essere considerati i vincoli sulla disponibilità dei dati di test, come la necessità di dati di test anonimizzati provenienti dalla produzione, o la creazione di dati di test specifici che possono essere difficili da fornire e che devono essere validati, come i dati per il testing dell'intelligenza artificiale (AI, Artificial intelligence). Ad esempio, il testing model-based può supportare la creazione e la gestione dei dati di test.

Il test manager dovrebbe determinare quale combinazione di tecniche di test, livelli di test e tipi di test dovrebbe essere utilizzata come approccio migliore per soddisfare la strategia di test dell'organizzazione, il contesto di progetto e fattori o vincoli aggiuntivi relativi al testing.

1.4.3 Definizione degli Obiettivi del Test

Un test plan dovrebbe essere definito per ogni progetto di test e dovrebbe contenere, tra gli altri aspetti, l'ambito del test, gli obiettivi del test e i criteri di uscita. Il test plan può essere impostato a livello di release, come un test plan di progetto (noto anche come master test plan) e, se necessario, come un level test plan per i diversi livelli di test. Inoltre, è possibile definire i test plan per le diverse caratteristiche di qualità, come un piano dei test di sicurezza o un piano per i performance test. Nello sviluppo software Agile e nello sviluppo software ibrido, è possibile concordare un iteration test plan. Per ogni release e iterazione, l'ambito delle caratteristiche funzionali e delle relative caratteristiche non-funzionali da rilasciare viene definito nel test plan e concordato con gli stakeholder.

In aggiunta alle funzionalità rilasciate per il testing in un progetto, devono essere definiti gli obiettivi del test e i criteri di uscita del progetto. Questo può essere fatto utilizzando la metodologia degli obiettivi S.M.A.R.T.:

- S = specific (specifico). Un obiettivo del test o un criterio di uscita di un progetto deve essere chiari e non ambigui.
- M = measurable (misurabile). Un obiettivo del test o un criterio di uscita di un progetto dovrebbe essere quantificabile e avere criteri specifici per misurare l'avanzamento e stabilire se è stato raggiunto.
- A = achievable (realizzabile). Un obiettivo del test o un criterio di uscita di un progetto dovrebbe essere fattibile considerando le risorse, i tempi e le capacità disponibili.
- R = relevant (rilevante). Un obiettivo del test o un criterio di uscita di un progetto dovrebbe essere allineato con gli obiettivi generali del progetto.
- T = timely (tempestivo). Un obiettivo del test o un criterio di uscita di un progetto dovrebbe avere una tempistica specifica e una scadenza definita per il completamento.

Gli obiettivi del test relativi al progetto dovrebbero riguardare tutti gli aspetti di qualità e quantità, purché siano misurabili o valutabili. Esempi di obiettivi del test di un progetto sono:

- Raggiungere i criteri di uscita specificati nel periodo temporale definito
- Soddisfare gli obiettivi di qualità dell'organizzazione (p.e. misurati come key performance indicator per il numero di reclami sul prodotto da parte dei clienti)
- Essere conforme alle normative e ai regolamenti del settore specifico
- Garantire la disponibilità dei dati solo agli utenti autorizzati (p.e. in base ai diritti di accesso)
- Verificare la completezza funzionale, la correttezza funzionale, le prestazioni, l'efficienza, la portabilità e la sicurezza della migrazione dei dati
- Aumentare il livello di test automation (p.e. aumentare i regression test o i performance test di una specifica percentuale)
- Eseguire il refactoring del codice con successo e dimostrare che non ha introdotto nuovi difetti (p.e. eliminare il codice sorgente poco strutturato o il debito tecnico mantenendo la funzionalità esistente, validata da un regression test)

- Dimostrare la sicurezza delle interfacce (p.e. validando i messaggi Extensible Markup Language (XML) rispetto alla Definizione dello Schema XML per garantire il rifiuto di dati non sicuri)
- Verificare l'usabilità di un'interfaccia utente e raggiungere un certo grado della sotto-caratteristica (p.e. misurando il tempo necessario per completare un compito specifico in un negozio online)

Oltre al conteggio e alla misurazione degli obiettivi del test di un progetto, dovrebbe essere presa in considerazione la valutazione del livello di qualità da parte degli esperti del dominio e degli stakeholder.

In base al contesto di progetto e agli obiettivi del test, a volte possono essere necessari più ambienti di test con le risorse e/o gli strumenti di test disponibili. Gli ambienti di test possono non essere tutti disponibili nello stesso momento. Questa necessità deve essere presa in considerazione quando si formulano obiettivi del test e criteri di uscita raggiungibili.

A seconda del contesto di progetto, si dovranno prendere in considerazione fattori aggiuntivi nel definire gli obiettivi del test e l'ambito del test di un progetto, come descritto nel paragrafo 1.2, *Il Contesto del Testing*.

1.5 Migliorare il Processo di Test

Introduzione

Il testing è una parte importante dello sviluppo del software e spesso rappresenta almeno il 30-40% dei costi totali del progetto. Oltre alle molte sfide (tecniche) che i progetti software devono affrontare (p.e. complessità e dimensioni crescenti, nuove tecnologie, ampia varietà di dispositivi e sistemi operativi, e vulnerabilità della sicurezza), esiste la necessità di ottimizzare l'efficacia e l'efficienza del testing e di conseguenza la necessità di migliorare i processi di test. Imparare dalle best practice esistenti e dai propri errori permette di migliorare il processo di test e di rendere i progetti più efficaci.

Un processo di miglioramento a livello di organizzazione è in genere più utile di un processo di miglioramento a livello di progetto o di team. Tuttavia, è possibile e vantaggioso applicare il process improvement (miglioramento del processo) a livello di progetto o di team, ma dovrebbe essere adattato alle esigenze del progetto o del team. Un miglioramento del test può essere avviato, ad esempio, dall'insoddisfazione per i risultati dei test attuali, da difetti inattesi, da circostanze mutevoli, da un risultato di benchmark o da una mancanza di comunicazione. Sono disponibili diverse tecniche per migliorare il testing (Bath & van Veenendaal, 2014). Alcune di queste tecniche sono descritte di seguito. Le tecniche descritte in questo Syllabus possono essere applicate sia ai modelli di sviluppo sequenziale sia ai modelli di sviluppo software Agile/incrementale. Il Syllabus ISTQB® Expert Level Improving the Test Process offre una visione più approfondita.

1.5.1 Il Processo di Miglioramento del Test (IDEAL)

Una volta che è stato concordato che i processi del test dovrebbero essere migliorati, le attività di implementazione del process improvement da adottare possono essere definite secondo le attività del modello IDEAL, che si basa su idee simili al noto ciclo plan-do-check-act (PDCA). IDEAL è acronimo di Initiating, Diagnosing, Establishing, Acting and Learning.

Sebbene IDEAL sia stato originariamente definito per supportare le attività di miglioramento a livello di organizzazione, può essere applicato anche a livello di progetto o a livello di team dello sviluppo software Agile. In un contesto di progetto, devono ancora essere raggiunti gli obiettivi delle attività (si veda sotto). La differenza principale è probabilmente la fase di Initiating, che è molto più ridotta a livello di progetto o di team rispetto al livello di organizzazione. La fase di Diagnosing attraverso una retrospettiva e la fase di Establishing di un plan sono molto probabilmente più ridotte rispetto al livello di organizzazione. Anche le fasi di Acting e Learning saranno rilevanti a livello di progetto o di team.

Initiating (Avvio) del process improvement

All'inizio del process improvement, gli obiettivi e l'ambito dei miglioramenti del processo vengono concordati dagli stakeholder.

Diagnosing (Diagnosi) della situazione attuale

Il processo di test attuale viene valutato per identificare i possibili miglioramenti. Nel caso di test process improvement (miglioramento del processo di test) model-based, la valutazione viene generalmente svolta rispetto a un framework standard (si veda il paragrafo 1.5.2, *Test Process Improvement Model-based*). Nel caso di test process improvement analitico, la valutazione può essere basata sull'analisi di metriche specifiche (si veda il paragrafo 1.5.3, *Approccio al Test Process Improvement Analitico*).

Establishing (Stabilire) un piano di test process improvement

Un piano di test process improvement può essere un documento formale che elenca tutte le azioni dettagliate che devono essere eseguite per ottenere miglioramenti. In base al contesto, il piano può essere molto informale e molto leggero. L'elenco dei possibili miglioramenti del processo dovrebbe essere prioritizzato. La prioritizzazione può essere basata sul Return On Investment (ROI, ritorno sull'investimento), sui rischi, sull'allineamento con le strategie del progetto o del team, e/o sui benefici quantitativi o qualitativi misurabili che devono essere raggiunti.

Acting (Agire) per implementare il test process improvement

Viene implementato il piano di test process improvement per il rilascio dei miglioramenti. Questo generalmente include la formazione e il piloting dei processi modificati, e la loro distribuzione completa nel progetto o nel team.

Learning (Apprendere) dal programma di miglioramento

Dopo aver distribuito tutti i miglioramenti del processo, è essenziale verificare quali benefici, pianificati o non previsti, sono stati ricevuti. Quando è stato appreso quello che ha funzionato e quello che non ha funzionato, è necessario agire sulla base di queste informazioni e solo in seguito si può avviare il successivo ciclo di miglioramento.

1.5.2 Test Process Improvement Model-based

Una premessa sia per il test process improvement model-based sia per il test process improvement analitico è l'assunzione che la qualità del prodotto è fortemente influenzata dalla qualità dei processi utilizzati e applicati. Quando si applica il test process improvement model-based, si utilizza un modello di miglioramento del test. I modelli di miglioramento del test si basano sulle best practice del testing e organizzano il miglioramento del test in modo graduale.

Sono emersi diversi modelli di processo raccomandati che supportano il test process improvement. Questi includono il Test Maturity Model integration (TMMi®) e TPI NEXT®.

Il miglioramento model-based può essere applicato anche a livello di progetto. In questi casi, la valutazione e il processo di miglioramento si focalizzano specificamente sui processi di test o sulle aree chiave definite nel modello, che sono relative alle attività a livello di progetto (p.e. pianificazione dei test e progettazione dei test) e che spesso omettono le attività a livello di organizzazione (p.e. politica di test e organizzazione del test). In alternativa, è possibile anche adattare in modo appropriato le pratiche che indirizzano il livello di organizzazione al contesto di progetto.

Per ulteriori informazioni sul test process improvement model-based si veda il Syllabus ISTQB® Expert Level Improving the Test Process.

Test Maturity Model integration

TMMi® (van Veenendaal & Cannegieter, 2011) (van Veenendaal, 2020) è composto da cinque livelli di maturità. Ogni livello di maturità, tranne il livello 1, contiene aree del processo di test e obiettivi di miglioramento. Inoltre, per facilitare e supportare la sua implementazione, TMMi® contiene pratiche, sottopratiche ed esempi. Il TMMi® è stato inizialmente sviluppato come complementare al Capability Maturity Model Integration (CMMI®), ma oggi è ampiamente utilizzato in modo indipendente dal CMMI®.

Per facilitare e supportare l'aggiornamento di TMMi® nello sviluppo software Agile, è stata sviluppata una linea guida specifica che spiega come TMMi® possa essere utilizzato e applicato in modo vantaggioso nello sviluppo software Agile.

Per ulteriori informazioni su TMMi®, si veda www.tmmi.org.

TPI NEXT®

Il modello TPI NEXT® (van Ewijk, 2013) definisce 16 aree chiave, ognuna delle quali copre un aspetto specifico del processo di test (p.e. strategia di test, metriche di test, strumenti di test e ambiente di test). Il modello definisce quattro livelli di maturità per ognuna delle 16 aree chiave.

Vengono definiti checkpoint specifici per valutare ogni area chiave a ogni livello di maturità. I risultati della valutazione vengono riassunti e visualizzati attraverso una matrice di maturità che copre tutte le aree chiave.

Per ulteriori informazioni su TPI NEXT® si veda www.tmap.net.

1.5.3 Approccio al Test Process Improvement Analitico

Utilizzando un approccio di miglioramento model-based, come descritto nel paragrafo precedente, i miglioramenti vengono introdotti confrontando l'approccio del test di un progetto o di un team con le best practice esterne. Gli approcci analitici identificano i problemi sulla base dei dati del progetto o del team stesso. Dall'analisi di questi problemi possono essere derivati i miglioramenti appropriati. Gli approcci analitici possono essere utilizzati insieme a un approccio model-based per verificare i risultati e fornire le differenze.

I problemi possono essere identificati utilizzando dati quantitativi e qualitativi. Questo paragrafo introduce approcci analitici che valutano l'approccio attuale utilizzando principalmente i dati quantitativi del processo di test e i dati dei difetti. Il paragrafo 1.5.4, *Retrospective*, introduce le retrospective, in cui i membri del team di sviluppo e di test raccolgono dati qualitativi su ciò che funziona bene e ciò che non funziona bene.

L'analisi dei dati è importante per il test process improvement oggettivo ed è un valido supporto alle valutazioni puramente qualitative, che altrimenti possono portare a raccomandazioni imprecise che non sono supportate dai dati. L'applicazione di un approccio analitico al miglioramento comporta il più delle volte un'analisi quantitativa del processo di test, per identificare le aree problematiche e stabilire gli obiettivi specifici del progetto. La definizione e la misurazione di parametri chiave vengono richieste per valutare il processo di test e stabilire se i miglioramenti hanno avuto successo.

Esempi di approcci analitici sono:

- Root cause analysis
- Analisi utilizzando misure, metriche e indicatori
- Approccio GQM (Goal-Question-Metric)

La root cause analysis è lo studio dei problemi per identificarne le relative root cause. Questo permette di identificare soluzioni che eliminano le cause dei problemi, piuttosto che limitarsi ad indirizzare i sintomi immediati ed evidenti. Una possibile procedura di analisi prevede di selezionare un insieme appropriato di difetti, di identificare cluster in questi dati, e di utilizzare diagrammi causa-effetto (chiamati anche diagrammi di Ishikawa o fishbone diagram) per identificare le root cause di importanti cluster di difetti. Sono quindi derivati dei miglioramenti per prevenire che si verifichino difetti simili.

Misure, metriche e indicatori sono utilizzati in modo quantitativo per valutare l'efficacia del processo di test eseguito nel progetto o nel team. Gli attributi chiave del processo di test da considerare sono l'efficacia, l'efficienza e la predicibilità. Per ognuno di questi attributi si possono selezionare una o più metriche.

Raccogliendo e analizzando i dati corrispondenti, possono essere identificate le aree chiave che richiedono un miglioramento.

L'approccio GQM (Goal-Question-Metric) (Basili, et al., 2014) (van Solingen & Berghout, 1999) fornisce un framework per la definizione e l'analisi di metriche, che vengono adattate alle esigenze informative degli stakeholder rilevanti del progetto. Gli obiettivi di misurazione definiscono un aspetto della qualità di un oggetto, che deve essere misurato per un particolare scopo, prospettiva e contesto. Questi obiettivi vengono raffinati in domande che definiscono l'aspetto della qualità dal punto di vista degli stakeholder. Vengono quindi selezionate le metriche che forniscono le informazioni necessarie per rispondere alle domande. I dati raccolti per le metriche rispondono alle domande, per valutare l'obiettivo di misurazione e soddisfare le esigenze informative degli stakeholder.

Ulteriori informazioni su questi approcci di test process improvement analitico si possono trovare nel Syllabus ISTQB® Expert Level Improving the Test Process.

1.5.4 Retrospective

Le retrospective sono meeting in cui un team esegue la review dei suoi metodi e della sua collaborazione, cattura le lessons learned (positive e negative) e decide sulle modifiche e azioni che devono essere attuate per ottenere miglioramenti (sia per i problemi del testing che per quelli non del testing). Le retrospective indirizzano argomenti quali il processo, le persone, l'organizzazione, la collaborazione e gli strumenti.

Le retrospective sono utilizzate sia nei modelli di sviluppo sequenziali sia nello sviluppo software Agile. Nei modelli di sviluppo sequenziali sono una parte del completamento dei test. In questo contesto, le retrospective hanno lo scopo di generare lessons learned per gestire meglio i progetti futuri. Nello sviluppo software Agile, le retrospective sono generalmente svolte alla fine di ogni iterazione per discutere che cosa ha avuto successo, che cosa richiede di essere migliorato, e come questi miglioramenti possano essere incorporati nell'iterazione successiva. Le retrospective vengono eseguite da tutto il team e quindi supportano l'approccio whole-team e favoriscono il continuous improvement (miglioramento continuo). Si noti che a volte sono necessarie retrospective dedicate per indirizzare i problemi del testing.

Una tipica retrospectiva consiste nei seguenti passi:

Introduzione: Viene svolta la review dell'obiettivo e dell'agenda della retrospectiva, creando un'atmosfera di fiducia reciproca in modo tale che i problemi possano essere discussi senza attribuire colpe o giudizi.

Raccogliere i dati: Vengono raccolti dati su quello che è accaduto durante l'iterazione o il progetto. È possibile raccogliere dati qualitativi, come ad esempio una timeline degli eventi chiave in modo da identificare i problemi ed elencare le opinioni di ogni membro del team in merito a tali problemi. Inoltre, possono essere presentati dati quantitativi provenienti da metriche. Ad esempio, i dati relativi all'avanzamento del test, al rilevamento dei difetti, all'efficacia del test, all'efficienza del test e alla predicibilità possono fornire una visione oggettiva sul testing del progetto o dell'iterazione.

Derivare i miglioramenti (improvement): I dati raccolti vengono analizzati per comprendere la situazione attuale e generare idee di miglioramento. Ad esempio, può essere applicata la root cause analysis per identificare le root cause dei problemi identificati, e può essere organizzata una sessione di brainstorming per generare idee su come risolverle.

Decidere le azioni di miglioramento: Vengono derivate e priorizzate le azioni per implementare le idee di miglioramento. Vengono definiti un piano di miglioramento e le responsabilità. Possono essere definiti obiettivi e metriche associate per valutare l'impatto delle azioni sui problemi identificati. Implementare troppi miglioramenti contemporaneamente è difficile da gestire con passaggi verificabili.

Chiudere la retrospettiva: In questo ultimo passo, la retrospettiva stessa viene sottoposta a review per identificare i punti di forza e di miglioramenti del processo di retrospettiva. La retrospettiva viene eseguita regolarmente, soprattutto nello sviluppo software Agile. Il continuous improvement (miglioramento continuo) viene applicato anche alla retrospettiva stessa.

È importante documentare in modo appropriato i risultati di una retrospettiva. In un modello di sviluppo sequenziale, i risultati, le conclusioni e le raccomandazioni devono essere distribuiti e comunicati in modo comprensibile ai membri dell'organizzazione. Nello sviluppo software Agile, i problemi e le azioni dovrebbero anche essere documentati per consentire la review delle azioni e del loro potenziale impatto sui problemi nell'iterazione successiva.

I tester, essendo parte del team (di progetto), portano la loro prospettiva che è unica. Possono sollevare problemi correlati al testing (e ad altro), e stimolare il team a pensare su possibili miglioramenti.

Ulteriori informazioni sono disponibili in (Derby & Larsen, 2006).

1.6 Strumenti di Test

Introduzione

Esistono tre tipi di strumenti di business:

- Strumenti commerciali
- Strumenti open-source
- Strumenti personalizzati (custom)

Quando si seleziona uno strumento di business, è necessario tenere conto di tutti i requisiti e di tutte le normative dell'organizzazione e degli stakeholder.

Esistono anche strumenti tecnici come gli strumenti di test automation, gli strumenti di test management e molti altri.

Esempi di utilizzo degli strumenti di test possono essere trovati nel Syllabus ISTQB® Foundation Level V.4.

1.6.1 Buone Pratiche per l'Introduzione di uno Strumento

Questo paragrafo contiene i passi necessari per la valutazione e l'introduzione di uno strumento di test.

Un test manager può essere coinvolto nell'introduzione di uno strumento, oppure può favorire o facilitare il processo di introduzione. I test manager sono generalmente responsabili di uno strumento di test dedicato, o di qualsiasi strumento correlato al testing, come uno strumento di requirements management, di defect management o di monitoraggio.

Esistono buone pratiche e considerazioni generiche per la valutazione e la selezione di uno strumento di test. Queste pratiche e considerazioni includono:

- Identificare le opportunità di process improvement, con il supporto di strumenti adeguati
- Comprendere le tecnologie utilizzate in un'organizzazione e selezionare uno strumento compatibile con queste tecnologie
- Comprendere come uno strumento sia tecnicamente e organizzativamente integrato nel SDLC
- Valutare lo strumento rispetto a requisiti chiari e criteri oggettivi
- Valutare il fornitore, nel caso si intenda utilizzare uno strumento commerciale. Valutare il supporto per gli strumenti non-commerciali (p.e. open-source)
- Identificare i requisiti interni per il coaching, il mentoring o la formazione sull'utilizzo dello strumento
- Considerare i pro e i contro dei vari modelli di licenza
- Come passo finale, eseguire una valutazione proof-of-concept

Le buone pratiche generiche per l'adozione e il rilascio di uno strumento includono:

- Eseguire un progetto pilota per validare i criteri di selezione e i requisiti, e per valutare come lo strumento si adatta ai processi e alle pratiche esistenti
- Adattare e migliorare i processi per adeguarli all'uso dello strumento, oltre ad adattare lo strumento ai processi esistenti, se necessario

- Definire le linee guida per l'utilizzo dello strumento
- Fornire formazione, coaching e mentoring agli utenti dello strumento
- Rilasciare lo strumento all'interno dell'organizzazione in modo incrementale
- Implementare un modo per raccogliere informazioni dall'utilizzo effettivo dello strumento, per ulteriori miglioramenti
- Definire chi è responsabile (ownership) dello strumento

1.6.2 Aspetti Tecnici e di Business per le Decisioni sugli Strumenti

La decisione di implementare e utilizzare uno strumento è influenzata da molteplici fattori. Per un test manager è importante conoscerli e indirizzarli.

- **Normative e sicurezza:** Le organizzazioni che sviluppano software safety-critical o mission-critical, o che sono soggette a conformità alle normative, possono preferire gli strumenti commerciali, in quanto soddisfano più spesso gli standard richiesti e possiedono spesso la certificazione appropriata.
- **Aspetti finanziari:** Gli strumenti open-source hanno generalmente un costo iniziale più basso grazie al supporto e allo sviluppo da parte della comunità. Gli strumenti commerciali possono avere un prezzo di acquisto una tantum oltre a costi di licenza ricorrenti. Il costo iniziale di uno strumento personalizzato (custom) è difficile da determinare perché dipende dai requisiti e dalla fase di sviluppo dello strumento. Oltre ai costi iniziali, devono essere calcolati e considerati i costi di formazione e di manutenzione per tutto il ciclo di vita dello strumento. Tutti gli strumenti possono avere alti costi di manutenzione e supporto.
- **Requisiti degli stakeholder:** È importante raccogliere i requisiti di tutti gli stakeholder per valutare e identificare lo strumento più appropriato. Gli strumenti commerciali e gli strumenti open-source non soddisfano necessariamente tutti i requisiti nel dettaglio. Gli strumenti personalizzati possono essere la scelta migliore per soddisfare tutti i singoli requisiti e nei casi in cui nessun altro strumento fornisca la funzionalità richiesta.
- **Landscape del software esistente e strategia degli strumenti:** È necessario valutare la composizione degli strumenti esistenti (landscape del software) e la relativa strategia degli strumenti, perché possono esistere fornitori preferiti o bloccati, sistemi integrati che hanno dipendenze con altri prodotti, oppure un modello speciale di supporto full-service per l'intero landscape del software con specifiche normative .

1.6.3 Considerazioni sul Processo di Selezione e Valutazione del Return on Investment (ROI, Ritorno sull'Investimento)

Gli strumenti di test possono rappresentare un investimento a lungo termine, che può estendersi su molte iterazioni di un singolo progetto, e/o essere applicabili a molti progetti. Un potenziale strumento deve quindi essere considerato da diversi punti di vista.

- Per il senior management, è necessario un Return on Investment (ROI, Ritorno sull'Investimento) positivo.
- Per il team di supporto e di operations, è preferibile un numero limitato ma necessario di strumenti utilizzati dall'organizzazione. Mantenere un numero più alto di strumenti, tenere traccia delle loro licenze e gestire l'insieme degli strumenti non dovrebbe richiedere costi o tempi elevati.

- Per i project leader, lo strumento deve aggiungere un valore misurabile al progetto o all'organizzazione.
- Per le persone che utilizzano lo strumento, è molto importante l'usabilità. L'usabilità include, ad esempio, il supporto per determinati compiti, l'apprendibilità e l'operatività.
- Per i membri del personale operativo, è importante la manutenibilità.

Le funzionalità devono essere analizzate per ogni tipo di strumento commerciale e tecnico. L'analisi è influenzata da diverse prospettive e interessi: test management, analisi (tecnica) del test, test automation o sviluppo. La persona dell'organizzazione responsabile dello strumento (owner dello strumento) deve assicurarsi che l'analisi sia portata a termine e che vengano presi in considerazione i punti sopra citati.

Tutti gli strumenti introdotti nel processo di test dovrebbero anche garantire un ROI positivo per l'organizzazione. È responsabilità del test manager occuparsi del calcolo e dell'ulteriore valutazione del ROI. Nello sviluppo software Agile può essere responsabilità dell'intero team di sviluppo.

Prima di acquistare o implementare uno strumento, dovrebbe essere eseguita un'analisi costi-benefici per garantire che sia vantaggioso per l'organizzazione. Questa analisi dovrebbe prendere in considerazione sia i costi ricorrenti sia i costi non-ricorrenti.

Le attività e i costi non-ricorrenti includono quanto segue:

- Definire e determinare i requisiti degli strumenti per il raggiungimento degli obiettivi
- Valutare e selezionare lo strumento e il fornitore corretti, eseguire un proof-of-concept
- Acquistare, adattare o sviluppare lo strumento per l'uso iniziale
- Definire le linee guida per l'utilizzo dello strumento
- Eseguire la formazione iniziale sullo strumento
- Integrare lo strumento nel landscape degli strumenti esistenti
- Procurare l'hardware/software necessario per supportare lo strumento

Le attività e i costi ricorrenti includono quanto segue:

- Costi di licenza e supporto ricorrenti
- Costi di manutenzione
- Costi di formazione continua
- Installazione dello strumento in ambienti diversi

Occorre inoltre considerare i costi di opportunità (opportunity cost). Questo significa che il tempo speso per la valutazione, la gestione, la formazione e l'utilizzo dello strumento avrebbe potuto essere impiegato per le attività di test effettive. Pertanto, potrebbero essere necessarie ulteriori risorse del test prima che lo strumento possa essere utilizzato per le attività previste.

Nella scelta degli strumenti dovrebbero essere considerati i seguenti rischi relativi al ROI:

- Immaturità dell'organizzazione che può portare a un uso inefficiente dello strumento
- Cambiamenti nella politica di manutenzione del fornitore che possono aumentare il carico di lavoro
- Costi più elevati rispetto a quanto previsto

- Beneficio inferiore rispetto a quanto previsto

I seguenti vantaggi possono essere applicati agli strumenti di test:

- Riduzione del lavoro manuale ripetitivo (p.e. regression testing)
- Accelerazione dei tempi del ciclo di test attraverso l'automazione
- Riduzione dei costi di esecuzione dei test grazie alla diminuzione delle attività manuali
- Maggiore copertura per alcuni tipi di test supportati dallo strumento
- Riduzione dell'errore umano grazie al minor numero di attività manuali
- Accesso più rapido alle informazioni sul test

Ulteriori vantaggi e rischi, in particolare per gli strumenti di test automation, possono essere trovati nei Syllabi ISTQB® Foundation Level V.4 e ISTQB® Test Automation Engineer.

In generale, un'organizzazione del test raramente utilizza un unico strumento. Il ROI totale di un'organizzazione è di solito un mix del ROI di tutti gli strumenti utilizzati per il testing. Gli strumenti devono condividere le informazioni e lavorare in modo cooperativo. È consigliabile una strategia completa e a lungo termine per gli strumenti di test che includa rischi, costi e benefici.

1.6.4 Ciclo di Vita dello Strumento

Esistono quattro fasi differenti nel ciclo di vita di uno strumento. È necessario nominare un amministratore dello strumento per garantire che le attività di queste fasi siano definite, eseguite e gestite.

- **Acquisizione:** Prima di tutto, è stata presa la decisione di selezionare uno strumento. Nella seconda fase, è necessario assegnare un owner dello strumento. Questa persona prende decisioni sull'utilizzo dello strumento (p.e. naming convention dei prodotti di lavoro, dove memorizzare i prodotti di lavoro). Prendere queste decisioni in anticipo può fare una differenza significativa nell'eventuale ROI dello strumento.
- **Supporto e manutenzione:** L'owner dello strumento è accountable (prende decisioni) della manutenzione dello strumento. La responsabilità delle attività di manutenzione dovrebbe essere dell'amministratore dello strumento o di un team dedicato agli strumenti. In caso di interoperabilità, è necessario considerare lo scambio dati e i processi di cooperazione e comunicazione. Inoltre, è necessario prendere decisioni sul backup e restore degli artefatti relativi allo strumento.
- **Evoluzione:** Con il passare del tempo, l'ambiente, le esigenze di business o le decisioni dei fornitori possono richiedere modifiche allo strumento. Quanto più complesso diventa l'ambiente operativo di uno strumento, tanto più facilmente una modifica può interrompere il suo utilizzo.
- **Ritiro:** Alla fine del suo ciclo di vita, lo strumento dovrebbe essere ritirato. Nella maggior parte dei casi le funzionalità fornite dallo strumento saranno sostituite e i dati dovranno essere preservati e/o archiviati. Il ritiro può essere basato su una decisione del fornitore oppure perché si è giunti a un punto in cui i vantaggi e le opportunità di passare a un nuovo strumento superano i costi e i rischi.

1.6.5 Metriche degli Strumenti

Le metriche oggettive degli strumenti sono progettate e raccolte in base alle esigenze del team di test e degli altri stakeholder. La maggior parte degli strumenti di test acquisiscono dati preziosi in tempo reale e

riducono l'effort della raccolta dei dati. Questi dati vengono utilizzati per gestire l'effort complessivo del test e per identificare le aree da ottimizzare.

Diversi strumenti sono focalizzati sulla raccolta di differenti tipi di dati. Esempi includono:

- Gli strumenti di test management possono fornire una varietà di metriche differenti relative agli elementi di test disponibili, ai test, ai test pianificati, allo stato di esecuzione dei test attuali e superati (p.e. superati, falliti, differiti, bloccati o pianificati)
- Gli strumenti di requirements management forniscono la tracciabilità relativa alla copertura dei requisiti rispetto ai test case superati e falliti
- Gli strumenti di defect management possono fornire informazioni sui difetti come lo stato, la severità, la priorità e la densità dei difetti degli elementi di test. Altri dati preziosi, come la defect detection percentage, i livelli di test in cui vengono introdotti i difetti e il tempo di risoluzione dei difetti rilevati, aiutano a guidare il process improvement, ma non tutti possono essere forniti esclusivamente dallo strumento di defect management
- Gli strumenti di analisi statica, tra gli altri, forniscono metriche relative alla complessità del codice
- Gli strumenti di performance testing possono fornire informazioni preziose come i tempi di risposta e i tassi di failure in caso di picchi di carico
- Gli strumenti di copertura del codice aiutano a comprendere quali parti dell'oggetto di test sono state esercitate dal testing
- Sebbene gli strumenti di test possano essere utilizzati per raccogliere metriche, dovrebbero anche essere utilizzati per il monitoraggio degli strumenti stessi. In questo contesto può essere misurata la qualità del processo di test (p.e. il numero di difetti rilevati con e senza strumenti, la copertura dei requisiti)
- Efficienza del test (p.e. durata dell'esecuzione dei test e numero di test eseguiti)

Maggiori dettagli sulla raccolta e sull'utilizzo delle metriche sono disponibili nel paragrafo 2.1, *Metriche di Test*.

2 Gestire il Prodotto - 390 minuti

Parole Chiave

anomalia, avanzamento del test, defect report, difetto, failure, metrica, obiettivo del test, stima del test, workflow del difetto

Parole Chiave Specifiche del Dominio

planning poker, three-point estimation, Wideband Delphi

Obiettivi di Apprendimento per il Capitolo 2:

2.1 Metriche di Test

- TM-2.1.1 (K2) Fornire esempi di metriche per raggiungere gli obiettivi del test
- TM-2.1.2 (K2) Spiegare come controllare l'avanzamento del test utilizzando le metriche di test
- TM-2.1.3 (K4) Analizzare i risultati del test per creare test report che consentano agli stakeholder di prendere decisioni

2.2 Stima del Test

- TM-2.2.1 (K2) Spiegare i fattori che devono essere considerati nella stima del test
- TM-2.2.2 (K2) Fornire esempi di fattori che possono influenzare le stime del test
- TM-2.2.3 (K4) Selezionare una tecnica o un approccio appropriati per la stima del test in un determinato contesto

2.3 Defect Management

- TM-2.3.1 (K3) Implementare un processo di defect management, includendo il workflow del difetto, che può essere utilizzato per monitorare e controllare i difetti
- TM-2.3.2 (K2) Spiegare il processo e i partecipanti necessari per un defect management efficace
- TM-2.3.3 (K2) Spiegare le specifiche del defect management nello sviluppo software Agile
- TM-2.3.4 (K2) Spiegare le sfide del defect management nello sviluppo software ibrido
- TM-2.3.5 (K3) Utilizzare i dati e le informazioni di classificazione che dovrebbero essere raccolti durante il defect management
- TM-2.3.6 (K2) Spiegare come le statistiche del defect report possono essere utilizzate per sviluppare il process improvement (miglioramento del processo)

2.1 Metriche di Test

Introduzione - Perché Avere delle Metriche di Test?

Il management utilizza questa frase: "Quello che viene misurato, viene fatto". Allo stesso modo, quello che non viene misurato è probabile che non venga fatto, perché è facile da ignorare. Pertanto, è importante stabilire un insieme appropriato di metriche per qualsiasi attività, incluso il testing.

Gli obiettivi del test sono la risposta al perché eseguiamo i test (si veda il paragrafo 1.4, *La Strategia di Test del Progetto*). Per determinare se gli obiettivi del test sono stati raggiunti, deve essere definito un modo per misurarli. Le metriche di test sono gli indicatori che ci aiutano a rispondere a questa domanda.

Le metriche di test possono essere categorizzate come segue:

- **Metriche di progetto:** misurano l'avanzamento rispetto ai criteri di uscita del progetto esistenti, come la percentuale dei test eseguiti, superati e falliti
- **Metriche di prodotto:** misurano gli attributi del prodotto, come il grado di soddisfazione del prodotto rispetto alle aspettative di qualità degli utenti a cui è destinato
- **Metriche di processo:** misurano la capacità del processo del testing e l'efficacia del testing. Sono quindi utilizzate per il reporting dell'efficacia e dell'efficienza relative al processo

Ulteriori informazioni sulla gestione delle metriche di prodotto e di processo sono presenti nel Syllabus ISTQB® Expert Test Management.

Ulteriori informazioni sull'uso delle metriche di processo possono essere trovate nel Syllabus ISTQB® Expert Level Improving the Test Process.

Nei paragrafi seguenti verranno discusse le metriche per la pianificazione dei test, il monitoraggio dei test, il controllo dei test e il completamento dei test. Queste sono le quattro principali attività di test management relative alle metriche.

2.1.1 Metriche per le Attività di Test Management

Questo Syllabus si focalizza sulle seguenti attività generiche di test management:

- Pianificazione dei test
- Monitoraggio dei test e controllo dei test
- Completamento dei test (si veda il paragrafo 1.1, *Il Processo di Test*).

Il test management deve essere in grado di definire un insieme di metriche di test per il monitoraggio dei test, il controllo dei test e il completamento dei test come parte delle attività di pianificazione dei test. Ogni metrica deve essere definita, misurata, monitorata e distribuita.

Durante la pianificazione dei test, vengono definite metriche di test appropriate che corrispondono agli obiettivi del test della strategia di test del progetto.

Le metriche utilizzate durante il monitoraggio dei test e il controllo dei test possono essere differenti da quelle utilizzate durante il completamento dei test. Durante il monitoraggio dei test e il controllo dei test, le metriche sono relative all'avanzamento delle attività di test. Durante il completamento dei test, dovrebbero essere raggiunti gli obiettivi del test. Una o più metriche potrebbero essere combinate per misurare i criteri di uscita del test assegnati a determinati obiettivi del test.

La seguente tabella fornisce esempi di metriche (esistono molte altre metriche) utilizzate nelle attività di test management:

Metriche (pianificate/monitorate per determinate milestone)	Monitoraggio dei test e controllo dei test	Completamento dei test
Copertura dei requisiti	X	X
Copertura dei rischi di prodotto	X	X
Copertura del codice	X	
Stima effettiva (in ore) delle attività del testing rispetto alla stima pianificata	X	
Percentuale di test case eseguiti per stato (p.e. fallito, bloccato) rispetto ai test case pianificati	X	X
Numero cumulativo di difetti risolti rispetto al numero cumulativo di difetti	X	
Test case automatizzati effettivi rispetto ai test case automatizzati pianificati		X
Costi effettivi del testing rispetto ai costi pianificati	X	

Tabella 2: Esempi di Metriche Utilizzate nelle Attività di Test Management

La metrica utilizzata in una specifica attività di test è indicata nella tabella. Le metriche con una X in “Monitoraggio dei test e controllo dei test” vengono utilizzate principalmente per misurare l'avanzamento, e sono riportate nei test progress report (si veda il Syllabus ISTQB® Foundation Level V.4). Le metriche con una X in “Completamento dei test” vengono utilizzate principalmente per misurare il raggiungimento degli obiettivi del test e sono riportate nei test completion report (si veda il Syllabus ISTQB® Foundation Level V.4). Le metriche con una X in entrambe le colonne possono essere utilizzate per entrambe le attività di test.

Esistono anche metriche per monitorare l'efficacia del test (p.e. defect detection percentage (DDP)).

La metrica DDP è trattata nel Syllabus ISTQB® Expert Level Improving the Test Process, in particolare nel paragrafo “*Implementing Test Process Improvement*”

2.1.2 Monitoraggio, Controllo e Completamento

Le metriche del test sono indicatori che mostrano lo stato di avanzamento del test, e se i criteri di uscita o i relativi compiti del test sono stati soddisfatti.

Il monitoraggio dei test è l'attività di raccolta dei dati relativi al test, e della relativa valutazione. Viene utilizzato per valutare l'avanzamento del testing e per verificare il raggiungimento dei criteri di uscita o l'esecuzione delle attività di test associate (si veda il paragrafo 1.1.2, *Monitoraggio e Controllo dei Test*). I criteri di uscita sono derivati dagli obiettivi del test.

Il controllo dei test utilizza le informazioni provenienti dal monitoraggio dei test per fornire indicazioni e azioni correttive al fine di ottenere un testing efficace ed efficiente. Esempi di direttive di controllo dei test includono: riassegnare la priorità dei test quando un rischio identificato diventa un problema, rivalutare se

un elemento di test soddisfa i criteri di ingresso o i criteri di uscita a causa di un'attività di rework, adeguare la schedulazione dei test per tenere conto di un ritardo nella consegna dell'ambiente di test, e aggiungere nuove risorse quando e dove necessario.

Il completamento dei test raccoglie i dati delle attività di test completate per consolidare le lessons learned, il testware e altre informazioni rilevanti. Il completamento dei test avviene in specifiche milestone di progetto, come il completamento di un livello di test, il completamento di un'iterazione, il completamento (o la cancellazione) di un progetto di test, il rilascio di un prodotto o il completamento di una release di manutenzione.

Metriche di test comuni utilizzate nelle attività di test management includono: metriche di avanzamento del progetto, metriche che mostrano l'avanzamento rispetto alla schedulazione e al budget pianificati, la qualità attuale degli elementi di test e l'efficacia del testing rispetto agli obiettivi del test o agli obiettivi dell'iterazione.

2.1.3 Reporting del Test

Il test management dovrebbe comprendere come interpretare e utilizzare le metriche per comprendere e fornire report sullo stato del test. Per i livelli più alti del testing, come il testing di sistema, il testing di integrazione dei sistemi, il testing di accettazione e il testing di sicurezza, la base di test principale è generalmente costituita da prodotti di lavoro come le specifiche dei requisiti, gli use case, le user story e i rischi di prodotto. Le metriche di copertura strutturale sono più applicabili ai livelli più bassi del testing, come il testing di componente (p.e. la copertura delle istruzioni) e il testing di integrazione dei componenti (p.e. la copertura dell'interfaccia). Mentre il test management può utilizzare le metriche di copertura del codice per misurare l'estensione che i test esercitano sulla struttura del sistema sotto test, il reporting dei risultati del test di alto livello dovrebbe essere adattato al contesto e alle esigenze specifiche del progetto. Ad esempio, in ambienti che cambiano frequentemente, le metriche di copertura del codice possono essere utili per monitorare l'impatto delle modifiche al codice sulla test suite e identificare potenziali lacune o rischi. Inoltre, il test management dovrebbe comprendere che, anche se i test di componente e i test di integrazione dei componenti raggiungono il 100% della copertura strutturale, i difetti e i rischi di qualità devono essere indirizzati ai livelli di test più alti.

L'obiettivo delle metriche del reporting è fornire una comprensione immediata delle informazioni per il management. Le metriche possono essere riportate come una fotografia di una metrica in un determinato momento o come l'evoluzione di una metrica nel tempo per valutare la tendenza (trend).

I rischi di prodotto, i difetti, l'avanzamento del test, la copertura, i relativi costi ed effort del test vengono misurati e rappresentati in modi specifici alla fine del progetto.

Di seguito alcuni esempi di metriche che possono essere utilizzate per scopi diversi.

Le metriche relative ai rischi di prodotto includono:

- Percentuale dei rischi per cui tutti i test sono stati superati
- Percentuale dei rischi per cui alcuni o tutti i test sono falliti
- Percentuale dei rischi non ancora completamente testati

Queste metriche possono essere utilizzate per valutare la qualità della base di test e l'efficacia dei test case nella copertura dei rischi di prodotto.

Le metriche relative ai difetti includono:

- Numero cumulativo di difetti risolti rispetto al numero cumulativo di difetti rilevati

- Suddivisione del numero, o percentuale, di difetti classificati per:
 - Elementi di test o componenti
 - Sorgente del difetto (p.e. specifica del requisito, nuova funzionalità o regressione)
 - Rilasci del test
 - Livello di test o iterazione in cui è stato introdotto, rilevato e corretto
 - Priorità/severità
 - Root cause
 - Stato (p.e. rifiutato, duplicato, aperto, chiuso)

Queste metriche possono essere utilizzate per monitorare il processo di rilevamento e risoluzione dei difetti, identificare le aree ad alta densità di difetti o di alta severità dei difetti, valutare l'efficienza e l'efficacia del test.

Le metriche relative all'avanzamento del test includono:

- Stato di esecuzione dei test: numero totale di test pianificati, implementati, eseguiti, superati, falliti, bloccati e non eseguiti
- Effort del test: Numero di ore delle risorse effettive dedicate al testing rispetto a quelle pianificate

Le metriche relative alla copertura includono:

- Copertura dei requisiti: La percentuale di requisiti coperti dai test case
- Copertura dei rischi di prodotto: La percentuale dei rischi di prodotto identificati che sono stati mitigati dai test case
- Copertura del codice: La percentuale di istruzioni, rami, percorsi o condizioni del codice che vengono eseguiti dai test case

Le metriche relative ai costi e all'effort del test includono:

- Rischi residui per i componenti non testati: L'impatto potenziale e la probabilità di difetti nei componenti non testati
- Costo del test: Il costo del testing effettivo rispetto a quello pianificato

Inoltre, è utile combinare metriche di categorie differenti (p.e. una metrica che mostri la correlazione tra l'andamento dei difetti aperti e l'andamento dei test eseguiti, oppure una metrica che mostri la qualità della base di test in base al numero di difetti rilevati nei requisiti). Quando l'esecuzione dei test continua e vengono identificati sempre meno difetti, si può decidere di terminare i test. Questa decisione dovrebbe basarsi sul reporting delle metriche e sui criteri di uscita concordati.

2.2 Stima del Test

Introduzione

Esistono best practice di project management per la stima dell'ingegneria dei sistemi e del software, relative a tutti i tipi di risorse (p.e. costi, persone o tempi). La stima del test è l'applicazione di queste best practice al testing associato a un progetto o a un'attività di operations.

2.2.1 Stimare quali Attività Coinvolgeranno il Testing

La stima del test è un'attività di test management che stima i tempi, l'effort e i costi necessari per completare un'attività. La stima del test è uno dei compiti principali e importanti del test management.

Le caratteristiche principali della stima nel test management sono:

- **Effort:** è generalmente calcolato in ore persona o in story point necessari per completare le attività di test del progetto. Spesso, l'effort del test e la durata del test (tempi di elapsed) possono essere differenti, e il test management può avere bisogno di stimare la durata totale dell'attività. Quante ore persona occorreranno?
- **Tempi:** quanto è richiesto per completare il progetto. Il tempo è una risorsa critica in un progetto. La pianificazione dei test deve stimare l'effort del test in giorni calendario e in giorni lavorativi. Ogni progetto ha delle milestone e una deadline (scadenza) per il rilascio. Quanto tempo sarà necessario per completare il progetto di test?
- **Costi:** è il budget del progetto. Include le spese per le risorse, gli strumenti e l'infrastruttura del test. Quanto costerà il progetto di test?

Il testing è spesso un sotto-progetto all'interno di un progetto (di grandi dimensioni), talvolta distribuito su diverse sedi di test (p.e. test center). Per eseguire la stima del test, il primo passo è identificare i livelli di test, le attività di test e i compiti del test. Quindi, dividere il progetto del testing nelle sue principali attività di test (p.e. pianificazione dei test ed esecuzione dei test) all'interno del processo di test (vedi il Syllabus ISTQB® Foundation Level V.4). Nei progetti Agile le attività del testing sono spesso stimate all'interno dell'effort dello sviluppo, e non come valori separati. Il passo successivo è stimare l'effort del test necessario per completare i compiti o i prodotti di lavoro, e quali sono i costi previsti per tale effort.

Poiché il testing è una sotto-attività di un progetto, esistono sempre alcuni vincoli naturali del progetto che lo influenzano e che richiedono un compromesso, e non possiamo manipolare questi valori in modo arbitrario. Questo può essere visto nel quality management come triangolo tempi-costi-qualità. Nel project management, il triangolo tempi-costi-qualità comprende tre valori che sono interdipendenti, e questo significa che sono strettamente correlati tra loro e si influenzano a vicenda. Questa relazione è comunemente osservata negli scenari di progetto.

2.2.2 Fattori che Possono Influenzare l'Effort del Test

La stima dell'effort del test consiste nel prevedere la quantità di lavoro relativo alle attività di test che sarà necessario per soddisfare gli obiettivi del test per un particolare progetto, release o iterazione. I fattori che influenzano l'effort del test possono includere le caratteristiche relative a:

Prodotto:

- Qualità della base di test
- Dimensioni del prodotto da testare (cioè l'oggetto di test)
- Complessità del dominio di prodotto (p.e. ambiente, infrastruttura e storia)
- Requisiti per il testing delle caratteristiche di qualità (p.e. sicurezza e affidabilità)

Questi fattori relativi al prodotto possono influenzare le stime del test, perché creano un contesto specifico per il sistema sotto test.

Processo di sviluppo:

- Stabilità e maturità dei processi di sviluppo dell'organizzazione
- Modello di sviluppo in uso (p.e. sviluppo software Agile/iterativo o modello di sviluppo software ibrido)
- Fattori materiali (p.e. disponibilità della test automation, strumenti e ambienti di test)

Questi fattori relativi al processo di sviluppo possono influenzare le stime del test, perché il testing è direttamente correlato allo sviluppo.

Persone:

- Soddisfazione delle persone (p.e. disponibilità nei giorni festivi, durante le ferie, oppure altri benefici attesi)
- Competenze ed esperienza delle persone coinvolte, soprattutto per quanto riguarda progetti e prodotti simili (p.e. la conoscenza di dominio)

Le persone sono le risorse più necessarie, quindi dovrebbe essere presa in considerazione qualsiasi instabilità. Pertanto, le persone sono un fattore importante nella stima dell'effort del test. Si veda anche il paragrafo 3.1, *Il Team di Test*.

Risultati del test:

- Numero e severità dei difetti rilevati durante l'esecuzione dei test
- Quantità di rework richiesto

Dati e statistiche storiche supportano la stima del test. Pertanto, la conoscenza di questi fattori supporterà una stima con valori più accurati.

Contesto del test:

- Distribuzione del testing su diverse sedi, la composizione e distribuzione geografica dei team, la complessità del progetto (p.e. più sottosistemi)
- Tipo di lavoro (p.e. virtuale oppure on-site)

I fattori relativi al contesto sono quelli che influenzano la stima completa del test. Si veda il paragrafo 1.2, *Il Contesto del Testing*.

2.2.3 Selezione delle Tecniche di Stima del Test

La stima del test dovrebbe coprire tutte le attività coinvolte nel processo di test. I costi, l'effort e, soprattutto, la durata di esecuzione dei test che sono stati stimati sono spesso i valori più importanti per il test management, perché questi valori avranno un impatto sul progetto. Tuttavia, le stime di esecuzione dei test

tendono a essere difficili da generare quando la qualità complessiva del software è bassa o non conosciuta. Inoltre, la familiarità e l'esperienza con il prodotto probabilmente influenzeranno la qualità delle stime. Una pratica comune è stimare il numero di test case derivati dalla base di test (p.e. i requisiti o le user story). Le assunzioni fatte durante la stima del test dovrebbero essere sempre documentate come parte della stima.

Le tecniche o gli approcci di stima del test possono essere classificati come metrics-based o expert-based.

Ulteriori dettagli sulle tecniche di stima del test sono trattati nel Syllabus ISTQB® Foundation Level V.4.

Nella maggior parte dei casi, la stima, una volta preparata, deve essere rilasciata al project management, insieme a una giustificazione. Frequentemente, alcuni parametri di input vengono modificati (p.e. l'ambito del test) e questo comporta un adeguamento della stima. Idealmente, la stima del test finale rappresenta il miglior bilanciamento possibile tra gli obiettivi dell'organizzazione e del progetto in termini di qualità, schedulazione, budget e funzionalità.

È importante tenere presente che qualsiasi stima si basa sulle informazioni disponibili al momento della sua preparazione. Nelle fasi iniziali di un progetto, le informazioni possono essere piuttosto limitate. Inoltre, queste informazioni possono cambiare nel tempo. Per rimanere accurate, le stime dovrebbero essere aggiornate per riflettere informazioni nuove o che vengono modificate.

La scelta della tecnica di stima dipende da diversi fattori, quali:

- **Errore di stima:** Alcune tecniche forniscono un modo per calcolare la deviazione standard, che è una misura dell'incertezza o della variabilità della stima. Ad esempio, la tecnica di three-point estimation utilizza la stima ottimistica, la stima pessimistica e la stima più probabile per calcolare il valore atteso e la deviazione standard della stima (per maggiori dettagli, si veda il Syllabus ISTQB® Foundation Level V.4, paragrafo 5.1.4).
- **Disponibilità dei dati:** Alcune tecniche richiedono dati storici di progetti precedenti o simili che possono non essere disponibili o affidabili. Ad esempio, le stime basate su valori statistici e sull'estrapolazione fanno riferimento a dati storici per derivare i valori statistici o le tendenze per il progetto attuale.
- **Disponibilità degli esperti:** Alcune tecniche richiedono il coinvolgimento di esperti che abbiano la conoscenza e l'esperienza per fornire stime accurate e realistiche. Ad esempio, il metodo Delphi e il planning poker si basano sulle opinioni e sui giudizi di esperti o dei membri del team.
- **Conoscenza della modellazione:** Alcune tecniche richiedono l'uso di formule o modelli matematici per calcolare le stime, che possono richiedere alcune competenze e conoscenze nella modellazione. Ad esempio, l'estrapolazione e la three-point estimation utilizzano delle formule per derivare il valore atteso e la deviazione standard della stima.
- **Vincoli di tempo:** Alcune tecniche richiedono più tempo ed effort per essere eseguite rispetto ad altre, e questo può influenzare la fattibilità e l'idoneità della tecnica. Ad esempio, il planning poker è semplice da eseguire, mentre l'estrapolazione può essere più difficile.

Questo evidenzia che i criteri per la selezione di tecniche appropriate per la stima del test dipendono fortemente dal contesto del testing (p.e. SDLC, stakeholder, livelli di test e tipi di test utilizzati nel progetto) (si veda il paragrafo 1.2, *Il Contesto del Testing*). Il test manager deve essere in grado di coordinare e applicare le tecniche di stima del test (p.e. con diversi modelli di SDLC in un progetto su diverse sedi).

Ad esempio, per selezionare la tecnica di stima appropriata, viene innanzitutto determinata la complessità dell'argomento. Se la complessità risultasse bassa, potrebbero essere utilizzate tecniche metrics-based.

Se la complessità risultasse alta, potrebbero essere utilizzate tecniche basate expert-based. Se viene utilizzato un modello di sviluppo sequenziale, può essere utilizzata la tecnica di stima Wideband Delphi. Se viene utilizzato un modello di sviluppo software Agile, può essere utilizzato il planning poker.

2.3 Defect Management

Introduzione

Il Syllabus ISTQB® Foundation Level V.4 descrive le attività che iniziano dopo aver osservato i risultati effettivi che differiscono dai risultati attesi. Il Syllabus si riferisce a queste attività come defect management. Altri standard utilizzano il termine "incident management", ISO/IEC/IEEE 29119-3 Standard o "anomaly management" (TMAP) per sottolineare il fatto che all'inizio del processo potremmo non sapere se la discrepanza è causata da un difetto in un prodotto di lavoro o da qualcosa di differente (p.e. un failure della test automation o un'incomprensione dei requisiti da parte del tester). Il defect management e lo strumento utilizzato per gestire i difetti sono di importanza fondamentale per i tester e per gli altri membri del team coinvolti nello sviluppo del software. Le informazioni ricavate da un processo di defect management efficace consentono al team di test e agli altri stakeholder di progetto di ottenere una vista dello stato di un progetto durante tutto il SDLC. Il defect management è cruciale anche per decidere quali difetti saranno corretti. Questo garantisce che l'effort sia dedicato per lavorare con i difetti corretti. Se nel corso del tempo vengono raccolti e analizzati i dati relativi ai difetti, questo può aiutare a individuare le aree di potenziale miglioramento sia per il testing sia per gli altri processi all'interno del SDLC (p.e. una migliore prevenzione dei difetti grazie a un'architettura e a una progettazione tecnica migliorate).

Oltre a comprendere il ciclo di vita complessivo dei difetti e il modo in cui viene utilizzato per monitorare e controllare entrambi i processi di sviluppo e test del software, il test manager e i tester (o il team Agile nello sviluppo software Agile) devono anche essere a conoscenza dei dati che sono difficili da acquisire. Il test manager deve essere un sostenitore dell'uso corretto del processo di defect management e dello strumento di defect management selezionato.

2.3.1 Ciclo di Vita del Difetto

Ogni fase del SDLC dovrebbe includere attività per rilevare ed eliminare potenziali difetti. Ad esempio, le tecniche del testing statico (cioè le review e l'analisi statica) possono essere utilizzate sulle specifiche di progettazione, sulle specifiche dei requisiti e sul codice, prima di rilasciare questi prodotti di lavoro alle attività successive. Quanto prima i difetti saranno rilevati e corretti, tanto più basso sarà il costo complessivo della qualità per il prodotto. Il costo della qualità è ridotto al minimo quando ogni difetto viene corretto nella stessa fase in cui è stato introdotto (cioè, quando il processo software raggiunge un perfetto contenimento di fase).

Durante il testing statico vengono cercati i difetti. Durante il testing dinamico, la presenza di un difetto viene rivelata quando si genera un failure, che risulta in una discrepanza tra i risultati effettivi e i risultati attesi di un test (cioè un'anomalia). In alcuni casi, si verifica un risultato falso-negativo quando il tester non osserva l'anomalia. Quando si osserva un'anomalia, si dovrebbero eseguire ulteriori indagini. Questa indagine di solito inizia con l'apertura di un defect report secondo i processi di test management e defect management specificati. Un test fallito non sempre comporta la creazione di un defect report. Ad esempio, nel test-driven development, i test di componente, di solito automatizzati, sono utilizzati come forma di specifica di progettazione eseguibile. Finché lo sviluppo del componente non è completo, alcuni o tutti i test devono inizialmente fallire. Pertanto, il risultato di un test di questo tipo non è necessariamente causato da un difetto e di solito non viene tracciato attraverso un defect report.

Un defect report viene gestito attraverso un workflow (per semplicità e consistenza con la maggior parte degli strumenti di defect management, viene utilizzato di seguito il termine "workflow del difetto"), e si muove in una sequenza di stati del difetto. Nella maggior parte di questi stati, esiste una persona che copre il ruolo

(owner) di responsabile del defect report e di responsabile dell'esecuzione di un compito (p.e. analisi, correzione del difetto o test confermativo). Il seguente diagramma rappresenta un workflow del difetto:

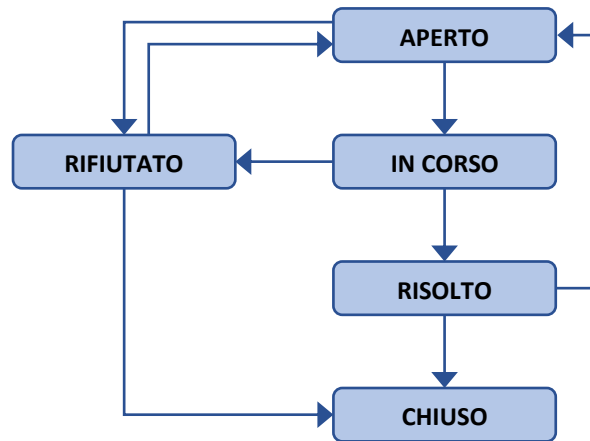


Figura 2: Un Semplice Workflow del Difetto

Un semplice workflow del difetto può coprire i seguenti stati del difetto:

- **APERTO** (può essere chiamato **NUOVO**): Lo stato iniziale quando viene creato il defect report.
- **IN CORSO**: Il team sta lavorando all'analisi e/o alla correzione del defect report.
- **RIFIUTATO**: Un defect report viene rifiutato dalla persona che lo ha analizzato (di solito uno sviluppatore o un analista). Possono esistere molte ragioni del rifiuto (p.e. informazioni non valide, test non corretto, defect report duplicato) e queste informazioni vengono aggiunte al defect report.
- **RISOLTO** (può essere chiamato **CORRETTO**, **PRONTO PER IL RETEST**): Un tester esegue un test confermativo, spesso seguendo i passi del defect report per riprodurre il failure, per determinare se la correzione ha effettivamente risolto il difetto.
- **CHIUSO**: Il defect report ha raggiunto lo stato finale e non è prevista l'esecuzione di ulteriore lavoro. Il tester seleziona questo stato per il defect report quando un test confermativo viene eseguito con successo oppure per confermare il rifiuto del defect report.

Un semplice workflow del difetto viene utilizzato in molte organizzazioni e può essere esteso con altri stati del difetto che sono rilevanti per un determinato contesto (p.e. **RIAPERTO**, **ACCETTATO**, **CHIARIMENTO RICHIESTO** o **RIMANDATO**).

Il workflow del difetto può essere differente in diverse organizzazioni, in termini di utilizzo di nomi diversi per gli stati del difetto, regole per le transizioni tra stati del difetto e ruoli responsabili dei compiti in determinati stati del difetto. Spesso il workflow del difetto è più semplice nello sviluppo software Agile piuttosto che nei modelli di sviluppo sequenziale. Il workflow del difetto dovrebbe essere adattato a un determinato contesto. Quando si progetta il workflow del difetto, è consigliabile rispettare alcune buone pratiche:

- Se possibile, il workflow del difetto dovrebbe essere definito a livello di organizzazione, per fornire un defect management unificato in tutti i progetti.

- I difetti duplicati e i falsi positivi dovrebbero essere rappresentati da uno stato separato o da una combinazione dello stato RIFIUTATO con la scelta del motivo del rifiuto. Possono essere utili per ulteriori analisi dei difetti allo scopo di migliorare il processo di test.
- Si raccomanda di utilizzare un solo stato finale (p.e. CHIUSO). La transizione a questo stato richiede spesso la scelta di un motivo per la chiusura, utile per le attività di valutazione e di process improvement.
- I nomi degli stati nel workflow del difetto dovrebbero essere gli stessi degli stati analoghi usati per altre entità (p.e. user story e compiti del test), in modo da semplificare il lavoro con questi.
- Gli stati consecutivi del difetto dovrebbero appartenere a differenti ruoli responsabili. Se due o più stati consecutivi appartengono allo stesso ruolo responsabile, dovrebbe esistere una buona ragione (p.e. per misurare il tempo trascorso in uno stato del difetto).
- Ogni stato del difetto, ad eccezione dello stato finale, dovrebbe avere più di una transizione in uscita, per consentire al ruolo responsabile di decidere il passo successivo. Le eccezioni a questa regola dovrebbero essere giustificate (p.e. per monitorare il tempo trascorso su una determinata attività).
- L'insieme degli attributi che viene richiesto di inserire quando si esegue una transizione di stato dovrebbe essere limitato a quelli che danno un valore sostanziale al defect management.

2.3.2 Defect Management Cross-funzionale

Sebbene l'organizzazione del test e il test manager siano spesso proprietari dell'intero processo di defect management e dello strumento di defect management, un team cross-funzionale è generalmente responsabile della gestione dei difetti per un determinato progetto. Questo team, talvolta chiamato comitato di defect management, può includere il test manager, i rappresentanti dello sviluppo, dei fornitori, del project management e del product management, il product owner e altri stakeholder che hanno un interesse nel software sotto test.

Man mano che le anomalie vengono rilevate e inserite nello strumento di defect management, il comitato di defect management dovrebbe stabilire se ogni defect report rappresenta un difetto valido e se dovrebbe essere risolto (e da quale team, nel caso in cui più team di sviluppo partecipino al rilascio), rifiutato o differito. Questa decisione richiede che il comitato di defect management consideri i benefici, i rischi e i costi associati alla correzione del difetto. È utile discutere tali considerazioni in un meeting (spesso chiamato meeting di triage). Se il difetto deve essere risolto, il team dovrebbe stabilire la priorità della risoluzione del difetto rispetto ad altri compiti. Il test manager e il team di test possono essere consultati in merito all'importanza relativa di un difetto e dovrebbero fornire le informazioni oggettive disponibili.

In progetti di dimensioni molto grande, nominare un defect manager full-time può essere giustificato dall'effort necessario per preparare e per seguire le decisioni prese nei meeting del comitato di defect management, almeno durante quelle fasi del SDLC dove il testing è più intensivo. In altre situazioni, diversi progetti di grandi dimensioni possono condividere un defect manager.

Uno strumento di defect management non dovrebbe essere utilizzato come sostituto di una buona comunicazione, e un comitato di defect management non dovrebbe essere utilizzato come sostituto di un uso efficace di un buon strumento di defect management. La comunicazione, l'adeguato supporto dello strumento, un workflow del difetto ben definito (che include le proprietà dei defect report) e un team di defect management coinvolto sono tutti elementi necessari per un defect management efficace ed efficiente.

2.3.3 Specifiche del Defect Management nei Team Agile

Il defect management nelle organizzazioni che utilizzano lo sviluppo software Agile è spesso lightweight (leggero) e/o meno formale rispetto ai modelli di sviluppo sequenziale. Se i team Agile sono co-locati oppure hanno a disposizione mezzi di comunicazione ben consolidati, le informazioni su un difetto o un failure spesso vengono scambiate tra i tester, i rappresentanti dei clienti e gli sviluppatori senza un defect report formale. Tuttavia, i defect report dovrebbero essere creati per:

- Difetti che bloccano altre attività dell'iterazione/sprint attuale (p.e. sviluppo, testing o altro) e che non possono essere risolti immediatamente all'interno del team Agile
- Difetti che non possono essere risolti all'interno della stessa iterazione. Alcuni team Agile hanno la regola di creare un defect report se il difetto non può essere risolto entro il giorno in cui il failure è stato rilevato
- Difetti che devono essere risolti da, o in collaborazione con, altri team in organizzazioni multi-team
- Difetti che devono essere risolti da un fornitore
- Difetti per i quali viene richiesto esplicitamente un defect report (p.e. quando uno sviluppatore non può lavorare immediatamente su una correzione)

La pratica comune è di aggiungere al product backlog i difetti che non possono essere risolti all'interno della stessa iterazione, in modo che possano essere prioritizzati rispetto agli altri difetti e alle user story per un'iterazione successiva.

Sebbene le basi del defect management dovrebbero essere impostate nella strategia di test di un'organizzazione, molti aspetti, come il livello di formalità, i trigger per la creazione di un defect report, e gli attributi dei difetti che devono essere catturati, possono essere lasciati da concordare tra i membri del team Agile. In generale, il livello di formalità del defect management e l'approccio per creare i defect report dovrebbero riflettere quanto segue:

- Membri del team co-locati
- Distribuzione dei membri del team tra diversi fusi orari
- Numero di team che collaborano allo sviluppo del prodotto
- Maturità del/dei team
- Dimensione del/dei team
- Rischi associati al prodotto
- Requisiti normativi, contrattuali o di altro tipo (se e dove applicabili)

La decisione finale del team Agile in merito ai dettagli del defect management dovrebbe essere sempre documentata (p.e. con linee guida in uno strumento di knowledge management).

2.3.4 Sfide del Defect Management nello Sviluppo Software Ibrido

Nella pratica, più team collaborano spesso al rilascio del sistema o del sistema di sistemi. Esempi sono lo sviluppo software ibrido, che si verifica quando un cliente usa lo sviluppo software Agile e uno dei suoi fornitori usa un modello di sviluppo sequenziale, oppure quando un'organizzazione che utilizza un modello di sviluppo sequenziale richiede la consegna di un sottosistema a un team che utilizza lo sviluppo software Agile. Questo ambiente multi-team pone diverse sfide:

- **Allineamento sugli attributi dei difetti e sugli strumenti da utilizzare per il defect management:** In uno scenario ideale, tutti i team utilizzano un unico strumento di defect management. Nella pratica, è comune che ogni team utilizzi uno strumento di defect management differente, soprattutto quando diversi team di fornitori diversi contribuiscono al rilascio del progetto. In questi casi è bene stabilire la sincronizzazione tra gli strumenti di defect management (preferibilmente in modo automatico).
- **Prioritizzazione dei difetti:** Il/i product owner dovrebbe/ro essere coinvolto/i nei meeting di defect management e cercare attivamente informazioni sulle conseguenze e sui rischi associati ai difetti. I meeting di defect management dovrebbero essere tenuti più spesso in uno sviluppo software Agile rispetto ai modelli di sviluppo sequenziale, per tenere il passo con la velocità di rilascio degli incrementi di prodotto da parte del team Agile. Tuttavia, questi meeting possono essere più brevi nei team Agile. A volte è utile che un gruppo più ristretto di stakeholder del defect management abbia l'ultima parola sulla prioritizzazione dei difetti.
- **Allineamento e trasparenza del test plan per il nuovo sviluppo e la correzione dei difetti:** Il lavoro di tutti i team dovrebbe essere allineato allo stesso piano di progetto, indipendentemente dal fatto che si stia utilizzando un modello di sviluppo software Agile o un modello di sviluppo sequenziale. Tutti i deliverable, comprese le correzioni dei difetti, dovrebbero essere allineati a questo piano di progetto. Un migliore allineamento può essere ottenuto con la partecipazione attiva dei membri di tutti i team al processo di pianificazione (p.e. la partecipazione dei team del modello di sviluppo sequenziale ai meeting dello sviluppo software Agile, dove i difetti vengono discussi e prioritizzati). La trasparenza dei piani di sviluppo può essere migliorata condividendoli tra i team (p.e. tramite dashboard o tramite il Product Backlog).

2.3.5 Informazioni del Defect Report

Le informazioni contenute in un defect report dovrebbero essere sufficienti per i seguenti scopi:

- Gestione del defect report attraverso il ciclo di vita del difetto
- Valutazione dello stato complessivo del progetto, soprattutto in termini di qualità del prodotto e di avanzamento del test
- Valutazione dello stato di un incremento di prodotto in termini di qualità del prodotto
- Valutazione della capacità del processo

Le informazioni necessarie per il defect management e per lo stato del progetto possono variare in base a quando il difetto viene rilevato nel SDLC. Inoltre, i defect report relativi a caratteristiche di qualità non-funzionali possono richiedere più informazioni (p.e. le condizioni di carico per i problemi relativi alle prestazioni). Tuttavia, le informazioni di base raccolte dovrebbero essere consistenti in tutto il SDLC e, idealmente, in tutti i progetti di un'organizzazione, per consentire un confronto significativo dei dati dei difetti nell'intero progetto e tra tutti i progetti.

Molti dati possono essere raccolti in un defect report. Il test manager dovrebbe decidere quali informazioni sono appropriate per un defect management efficace in un determinato contesto di progetto. Poiché ogni attributo aggiuntivo aumenta il tempo dedicato al reporting dei difetti e può aumentare la confusione della persona che sta inserendo il defect report, è consigliabile raccogliere solo i dati necessari per il defect management in un determinato contesto e/o che saranno utilizzati per il process improvement.

Per gestire il defect report nella maggior parte degli ambienti, sono obbligatori i seguenti elementi:

- Un titolo del difetto con una breve sintesi dell'anomalia

- Una descrizione dettagliata dell'anomalia, che preferibilmente includa i passi per riprodurre il failure
- Severità dell'impatto sul sistema sotto test e/o sugli stakeholder di prodotto
- Priorità per risolvere l'anomalia

Vengono spesso creati dati aggiuntivi importanti dallo strumento di defect management:

- Identificativo univoco per il defect report
- Data/ora di creazione del defect report
- Nome della persona che ha rilevato e/o documentato l'anomalia
- Progetto e fase del SDLC in cui è stata rilevata l'anomalia
- Stato attuale del defect report
- Owner attuale (cioè la persona responsabile e attualmente incaricata di lavorare sul difetto)
- Storia delle modifiche, come la sequenza di azioni, comprese le informazioni su data e ora, intraprese dai membri del team di progetto per isolare, correggere e confermare il difetto come corretto
- Riferimenti (p.e. al test case, ai difetti correlati).

In base al contesto, in un defect report possono essere raccolte anche altre informazioni (p.e. la tracciabilità) (per ulteriori informazioni, si veda lo standard ISO/IEC/IEEE 29119-3). I punti seguenti raggruppano le informazioni in base allo scopo previsto:

- **Per facilitare la risoluzione del difetto:** Il sottosistema o il componente in cui si trova il difetto, lo specifico elemento di test e il relativo release number in cui è stata osservata l'anomalia, oppure l'ambiente di test in cui è stato osservato il difetto
- **Per valutare lo stato complessivo del progetto:** Informazioni per monitorare l'avanzamento (p.e. rischi, costi, opportunità e benefici associati alla correzione o alla non-correzione del difetto, una descrizione di workaround disponibili, oppure i requisiti impattati dal difetto)
- **Per valutare lo stato di un incremento di prodotto in termini di qualità del prodotto:** Il tipo di difetto (solitamente corrispondente a una tassonomia dei difetti), il prodotto di lavoro in cui è stato introdotto il difetto, oppure la caratteristica/sotto-caratteristica di qualità impattata dal difetto
- **Per valutare la capacità del processo:** Informazioni per monitorare l'efficacia e l'efficienza dei processi di sviluppo (p.e. la fase del SDLC di introduzione, rilevamento e correzione del difetto, oppure la root cause del difetto)

2.3.6 Definire le Azioni di Process Improvement Utilizzando le Informazioni del Defect Report

Come discusso nel paragrafo 2.3.5 di questo Syllabus, *Informazioni del Defect Report*, i defect report possono essere utili per il monitoraggio e reporting dello stato del progetto. Mentre le implicazioni delle metriche sul processo di test sono indirizzate principalmente nel Syllabus Expert Test Management, nel Test Management Advanced Level, i test manager dovrebbero essere consapevoli del significato dei defect report per valutare la capacità dei processi di sviluppo software e di test.

Oltre alle informazioni sul monitoraggio dell'avanzamento del test menzionate nel paragrafo 2.1.2, *Monitoraggio, Controllo e Completamento*, e nel paragrafo 2.1.3, *Reporting del Test*, le informazioni sui

difetti dovrebbero supportare le iniziative di process improvement discusse durante le retrospettive. Esempi includono:

- Utilizzare le informazioni sulle fasi di introduzione, rilevamento e correzione dei difetti per valutare il contenimento di fase e/o per eseguire l'analisi dei costi della qualità, con l'obiettivo di suggerire modi per migliorare l'efficacia del rilevamento dei difetti in ogni fase e per minimizzare i costi associati ai difetti
- Utilizzare le informazioni sulla fase di introduzione per analizzare le fasi in cui viene introdotto il maggior numero di difetti, consentendo miglioramenti mirati per la prevenzione dei difetti
- Utilizzare le informazioni sulla root cause dei difetti per determinare le ragioni alla base dell'introduzione dei difetti, consentendo miglioramenti del processo che riducano il numero totale dei difetti
- Utilizzare le informazioni sulla localizzazione dei difetti per eseguire l'analisi dei cluster di difetti, per comprendere meglio i rischi tecnici (per il testing basato sul rischio) e per consentire il refactoring dei componenti più difettosi
- Utilizzare le informazioni sui difetti riaperti per valutare la qualità delle implementazioni di debugging
- Utilizzare le informazioni sui difetti duplicati e rifiutati per valutare la qualità della creazione dei defect report
- Consentire miglioramenti del processo che riducano il numero totale dei difetti introducendo misure proattive per evitare gli errori in anticipo

L'uso di metriche per valutare l'efficacia e l'efficienza del processo di test è discusso nel Syllabus Expert Test Management.

In alcuni casi, i team decidono di non tracciare i difetti rilevati durante alcune o tutte le fasi del SDLC. Sebbene questo avvenga spesso in nome dell'efficienza e per ridurre le spese generali del processo, questo riduce notevolmente la visibilità sulle capacità del processo di sviluppo e sul testing. Questo rende difficile realizzare gli esempi di miglioramenti suggeriti sopra a causa della mancanza di dati di supporto affidabili.

3 Gestire il Team - 225 minuti

Parole Chiave

costo della qualità, difetto, failure, failure esterno, failure interno, prevenzione dei difetti, valutazione (appraisal),

Obiettivi di Apprendimento per il Capitolo 3:

3.1 Il Team di Test

- TM-3.1.1 (K2) Fornire esempi delle competenze tipiche necessarie ai membri del team di test all'interno delle quattro aree di competenza
- TM-3.1.2 (K4) Analizzare un determinato contesto di progetto per determinare le competenze richieste ai membri del team di test
- TM-3.1.3 (K2) Spiegare le tecniche tipiche per la valutazione delle competenze dei membri del team di test
- TM-3.1.4 (K2) Distinguere tra gli approcci tipici per lo sviluppo delle competenze dei membri del team di test
- TM-3.1.5 (K2) Spiegare le competenze richieste per gestire un team di test
- TM-3.1.6 (K2) Fornire esempi di fattori motivanti e di fattori di igiene per i membri del team di test

3.2 Relazione con gli Stakeholder

- TM-3.2.1 (K2) Fornire esempi per ognuna delle quattro categorie che determinano il costo della qualità
- TM-3.2.2 (K3) Applicare un calcolo costi-benefici per stimare il valore aggiunto del testing per gli stakeholder

3.1 Il Team di Test

Introduzione

Ogni team che svolge attività e compiti del test è composto da individui con competenze differenti. Mentre in alcune organizzazioni i team sono auto-organizzati, in altre i test manager ingaggiano e sviluppano questi team. Il giusto mix di competenze¹ è un fattore critico per tutti i team che devono completare con successo i compiti del testing.

Le competenze richieste a un membro del team di test possono cambiare nel tempo. È importante selezionare le persone giuste per il team di test e fornire una formazione adeguata e opportunità di crescita. Inoltre, persone esterne al team di test possono fornire ulteriori competenze specifiche.

Questo capitolo esamina il processo fondamentale di analisi e sviluppo delle competenze richieste ai membri dei team di test, oltre alle competenze richieste per guidare o essere coach di un team di test. Questo include anche la conoscenza dei fattori che motivano o demotivano i membri del team di test e altri fattori che garantiscono il successo del lavoro di squadra.

Ogni individuo possiede già delle competenze e può svilupparle ulteriormente in diversi modi, ad esempio con l'esperienza lavorativa, l'istruzione e la formazione. Il team di test ideale possiede tutte le competenze necessarie per determinati compiti del test, oppure è responsabile solo dei compiti per i quali possiede le competenze richieste. Per avere successo, un team di test ha bisogno di diverse competenze a livelli differenti. In base al contesto di progetto, alcune competenze saranno più importanti o necessarie di altre. Può avere senso considerare esperti esterni per specifici compiti del testing che vanno oltre le capacità del team di test.

3.1.1 Competenze Tipiche all'Interno delle Quattro Aree di Competenza

Le competenze di una persona possono essere classificate in quattro aree di competenza (Sonntag, et al., 2005; Erpenbeck, et al., 2017)²:

- **Competenza professionale:** Include le competenze necessarie per eseguire compiti specializzati. Esempi includono competenze nelle tecniche di test, competenze tecnologiche e competenze di business nel dominio applicativo, nonché competenze di project management.
- **Competenza metodologica:** Include le competenze generali che una persona può utilizzare in modo indipendente in un dominio e che consentono l'esecuzione indipendente di compiti nuovi o complessi. Esempi includono capacità analitiche, concettuali e di giudizio.
- **Competenza sociale:** Include le competenze relative alla comunicazione, alla cooperazione e alla gestione del conflitto in contesti intra e interculturali. Consentono di relazionarsi agli altri per agire in modo appropriato in una determinata situazione e per raggiungere obiettivi individuali e condivisi.

¹ Il termine "competenza" è usato come termine 'ombrello' per le competenze stesse, per possedere la conoscenza di qualcosa e per l'abilità di fare qualcosa.

² Le quattro aree di competenza considerate si basano sul modello descritto in questi riferimenti, che è ampiamente utilizzato. Esistono altri modelli descritti nella letteratura che raggruppano le competenze in modo diverso. Questi altri modelli non sono parte di questo Syllabus.

Esempi includono competenze di comunicazione, competenze di risoluzione dei conflitti, capacità di lavorare in team, adattabilità e assertività.

- **Competenza personale:** Include l'abilità e la volontà di sviluppare sé stessi e il proprio talento, la motivazione e la volontà di ottenere buoni risultati, nonché lo sviluppo di attitudini specifiche e di una personalità individuale. Esempi includono autogestione, responsabilità personale, abilità nel ricevere critiche, affidabilità, resilienza, abilità di agire con fiducia, disciplina, apertura ai cambiamenti, disponibilità ad aiutare e ad imparare, e abilità di delegare.

Tutte le aree di competenza sono importanti per il successo di qualsiasi team del testing. Poiché le competenze metodologiche, sociali e personali non sono specifiche del testing, ISTQB® si focalizza sullo sviluppo delle competenze professionali. Questo include competenze nella gestione dei compiti del test, nell'analisi della base di test, nella progettazione dei test, nell'identificazione e analisi dei rischi e nello sviluppo, nella configurazione e nella manutenzione dei dati di test, degli ambienti di test e dei test script.

3.1.2 Analizzare le Competenze Richieste ai Membri del Team di Test

Definire il team di test è un'attività che rientra nella pianificazione dei test. Include il compito di identificare i ruoli e le competenze richieste dallo staff per implementare il testing nella strategia di test. Per determinare le competenze necessarie per un progetto è richiesta un'analisi dettagliata del contesto.

Competenza professionale e metodologica

Per il testing, il focus è sulle competenze del test richieste per i compiti del test. Di seguito sono riportati alcuni esempi:

- La pianificazione dei test richiede conoscenze concettuali per sviluppare una strategia di test.
- Il monitoraggio dei test e il controllo dei test richiedono competenze di project management per la gestione di tutti i compiti del test.
- L'analisi dei test richiede competenze analitiche per analizzare la base di test e i rischi di prodotto.
- La progettazione dei test richiede competenze nelle tecniche di test per progettare i test case, e una conoscenza concettuale per progettare gli ambienti di test.
- L'implementazione dei test richiede capacità di giudizio per la selezione dei test, e competenze tecniche per la programmazione dei test script e per l'installazione degli ambienti di test.
- L'esecuzione dei test richiede competenze tecniche per eseguire i test automatizzati, per eseguire il testing esplorativo e per valutare i risultati del test.
- Il completamento dei test richiede l'abilità di comunicare i risultati di progetto, e la responsabilità personale delle decisioni prese.

Differenti tipi di test e livelli di test richiedono competenze differenti (p.e. competenze di business nel dominio di applicazione per valutare l'adeguatezza funzionale di un sistema, o competenze tecniche per valutare la manutenibilità del codice).

Inoltre, il contesto di progetto fornisce informazioni preziose sulla competenza professionale richiesta:

- Il dominio di sistema richiede competenze di business nell'area applicativa, come l'information technology, l'industria automotive o l'industria del gioco d'azzardo (gambling).

- L'architettura software e di sistema, e le tecnologie utilizzate nel progetto richiedono, ad esempio, competenze tecniche nei linguaggi di programmazione, nella tecnologia dell'interfaccia o nelle vulnerabilità di sicurezza.
- Il SDLC richiede, ad esempio, la conoscenza dei livelli di test, dei ruoli del testing e delle tecniche di test specifiche.

Competenza sociale

Nel contesto del testing, la competenza sociale consente ai membri del team di test di comportarsi in modo appropriato nelle relazioni con gli altri membri del team e di raggiungere gli obiettivi del test. In particolare, include le competenze di comunicazione, cooperazione e risoluzione dei conflitti (p.e. gestire in modo costruttivo le condizioni di test non ottimali o eseguire il reporting dei difetti verso gli sviluppatori).

Lo sviluppo e il testing del software sono generalmente svolti da membri diversi in gruppi (differenti) che coordinano i loro compiti attraverso la comunicazione. Le competenze comunicative, la capacità di lavorare in team e l'abilità di risolvere i conflitti sono necessarie per il successo del progetto. Tuttavia, il livello richiesto di competenze sociali può variare in base al contesto di progetto. Ad esempio, lo sviluppo software Agile può richiedere più requisiti sulle competenze sociali rispetto ai modelli di sviluppo sequenziale basati sui documenti e al testing offsite.

Competenza personale

L'efficacia e l'efficienza dei membri del team di test dipende anche dalla loro capacità e volontà di sviluppare sé stessi, le loro competenze e le loro attitudini. Ad esempio, lavorare in un team Agile auto-organizzato può richiedere un livello più alto di autogestione e di disciplina da parte di tutti i membri del team, mentre un test manager di un team di test gerarchico deve, per esempio, essere in grado di delegare il lavoro. Spesso è richiesto un alto grado di affidabilità e resilienza, in particolare nei progetti time-critical. Inoltre, la disponibilità ad aiutare, ad imparare e l'apertura al cambiamento sono importanti in un processo di cambiamento in tutti i modelli del SDLC.

3.1.3 Valutare le Competenze dei Membri del Team di Test

In molti casi, i team di test vengono formati con lo staff esistente. Per comprendere le capacità dei membri del team e la necessità di sviluppo personale, il test management deve valutare le competenze esistenti del team di test e confrontarle con quelle richieste, che possono essere documentate in una matrice delle competenze.

Esistono modelli per aiutare i team e i membri dei team a lavorare in modo più efficace (p.e. "Team Roles" di Meredith Belbin, DISG® o PCM®). Secondo Belbin (Belbin, 2010) i team lavorano in modo efficace quando sono composti da diversi tipi di personalità e di ruoli. Questi modelli aiutano i team a identificare quali competenze hanno e quali competenze potrebbero mancare.

La competenza professionale e metodologica dei membri del team di test può essere valutata attraverso la dimostrazione dei compiti tipici del test:

- Delineare una strategia di test e discutere il feedback con i colleghi
- Eseguire la review della base di test e comunicare i risultati, che può rivelare anche competenze comunicative
- Determinare le tecniche di test per raggiungere obiettivi del test specifici per un determinato contesto di progetto

- Applicare varie tecniche di test in modo appropriato
- Scrivere un test completion report che includa una valutazione dei risultati del test

Inoltre, le competenze possono essere valutate attraverso credenziali, certificazioni, esperienze lavorative e titoli di studio.

Soprattutto nello sviluppo software Agile, i team identificano le competenze necessarie partecipando regolarmente alle retrospettive e ricevendo feedback. Coach o mentori esperti li aiutano a sviluppare le loro competenze, e a identificare e risolvere le lacune di conoscenza.

3.1.4 Sviluppare le Competenze dei Membri del Team di Test

All'inizio di un progetto, un team di test può non disporre di tutte le competenze richieste. Sebbene possa non essere disponibile un insieme perfetto di individui, un team forte può bilanciare i punti di forza e di debolezza dei singoli individui.

Il test management o il team di test possono identificare le esigenze di sviluppo necessarie, confrontando le competenze richieste con quelle disponibili in una matrice delle competenze. Su questa base, si possono determinare gli approcci allo sviluppo delle competenze:

- La formazione e l'istruzione insegnano conoscenze e pratiche predefinite, di solito in un'aula (virtuale) (p.e. mandando persone a un corso di formazione, organizzando sessioni di formazione in-house, eseguendo training custom oppure utilizzando corsi live di e-learning).
- Il self-study (auto-apprendimento) è una modalità di apprendimento di una materia che prevede lo studio da soli, piuttosto che in una classe (virtuale), (p.e. leggendo libri, guardando video registrati o ricercando risorse su Internet).
- Il peer learning (apprendimento tra pari) è una modalità in cui i colleghi condividono conoscenze, idee ed esperienze e imparano gli uni dagli altri.
- Il mentoring o il coaching sono approcci in cui un membro del team che è nuovo in un ruolo riceve una guida individuale da un coach, oppure conoscenze, competenze e/o esperienze da un mentore esperto. La persona esperta agisce come risorsa che in modo continuativo fornisce consigli e assistenza.
- Anche il training on the job è una modalità molto conosciuta e consiste in un mix di self-study, peer learning e mentoring o coaching.

Non tutti gli approcci allo sviluppo delle competenze sono ugualmente efficaci ed efficienti. Il self-study e la formazione, ad esempio, sono adatti a sviluppare competenze professionali e metodologiche. Per questo motivo, le conoscenze di base sul testing possono essere sviluppate partecipando alle sessioni di formazione ISTQB® o attraverso il self-study basato sui Syllabi ISTQB®. Tuttavia, per sviluppare le competenze sociali e personali, si raccomanda di utilizzare approcci come la formazione e il coaching, che spesso sono più promettenti del self-study. Lo scambio sociale, il feedback e la riflessione sono tra i fattori chiave di successo per lo sviluppo delle competenze sociali e personali.

3.1.5 Competenze di Management Necessarie per Gestire un Team di Test

Chiunque voglia guidare con successo un team di test deve possedere competenze di management. Queste includono la competenza professionale e metodologica nelle attività di management fondamentali (p.e. pianificazione, monitoraggio dell'avanzamento, controllo e reporting). Per il testing sono necessarie

conoscenze e competenze specifiche di test management (p.e. conoscenza di differenti approcci del test, competenza nello sviluppo di strategie di test, nell'utilizzo di tecniche di test o nel SDLC applicato).

Guidare o essere coach di un team di test significa agire in modo appropriato nelle relazioni con gli altri membri del team di test, e avere l'abilità e la volontà di svilupparsi in circostanze mutevoli. Per questo motivo, le competenze sociali e personali sono fattori di successo essenziali per guidare un team di test. Questo include la resilienza, l'abilità di delegare e la capacità di farsi accettare dal team di test. Inoltre, include la capacità di affermare gli interessi del test nel progetto, di promuovere i vantaggi del testing, e di comunicare e risolvere i conflitti con tutti gli stakeholder.

Per acquisire le persone da inserire nel team di test, sono necessarie capacità di analisi delle condizioni sociali, lavorative e del team. Queste competenze aiutano a garantire che il team di test si adatti alle condizioni di lavoro o, se possibile, che le condizioni lavorative siano adattate al team di test. Inoltre, i team di test sono soggetti a processi di sviluppo dinamici e quindi richiedono competenze situazionali (p.e. secondo le fasi del modello Tuckman per lo sviluppo di piccoli gruppi) (Tuckman, 1965; Bonebright, 2010):

- Disponibilità ad aiutare i membri del team di test ad unirsi al team di test (Forming)
- Capacità di risolvere i conflitti all'interno del team di test (Storming)
- Disciplina e leadership orientata agli obiettivi per garantire regole e valori concordati (Norming)
- Capacità di delegare per poter dare responsabilità individuali al team di test (Performing)
- Capacità di agire con apprezzamenti e fiducia nei confronti dei membri del team di test in uscita dal team (Adjourning)

3.1.6 Fattori Motivanti o Demotivanti per un Team di Test in Determinate Situazioni

Membri del team di test soddisfatti e motivati aumentano la produttività e le prestazioni, e quindi hanno un impatto significativo sul successo dei progetti. Quando si raggiunge questo obiettivo, il cross-training avviene in modo informale, i membri del team di test possono gestire il proprio carico di lavoro e il test management ha più tempo per occuparsi di problemi esterni. La teoria motivazionale di motivazione-igiene (Herzberg, et al., 1993) distingue tra fattori motivatori e fattori di igiene.

I fattori motivatori sono percepiti in modo consapevole e possono portare alla crescita e alla soddisfazione. Possono includere:

- Riconoscimento e apprezzamento per il lavoro svolto (p.e. incentivi e qualsiasi altro approccio individuale che i membri del team di test ritengono apprezzabile)
- Maggiore responsabilità e autonomia (p.e. per definire i processi di test in un team di test)
- Compiti interessanti, significativi e sfidanti che i membri del team di test percepiscono come raggiungibili e allo stesso tempo degni di essere perseguiti (p.e. la selezione e l'introduzione di un nuovo strumento per la test automation)
- Avanzamento e sviluppo professionale (p.e. un tester esperto che diventa test manager o responsabile del processo di test)

I fattori di igiene (del lavoro) sono di solito dati per scontati. Soddisfare questi fattori non porta automaticamente a una maggiore soddisfazione. Se mancanti, possono avere un effetto demotivante sui membri del team di test:

- Remunerazione appropriata (p.e. salario di mercato, straordinari retribuiti, buoni benefit sociali)

- Stile del management e politica del personale apprezzabili (p.e. gestione snella, obiettivi target realistici, protezione da accessi esterni e da sovraccarichi)
- Condizioni di lavoro piacevoli (p.e. specifiche non ambigue, oggetti del test maturi, difetti adeguatamente corretti, luogo di lavoro appropriato, ambiente di test stabile)
- Safety come bisogno esistenziale (p.e. posto di lavoro safe e aderenza agli agreement)
- Buone relazioni interpersonali (p.e. con colleghi e supervisor)

Di conseguenza, il test management dovrebbe continuamente eliminare i fattori demotivanti e allo stesso tempo creare e rafforzare i fattori motivanti.

Ulteriori informazioni sono disponibili in (Belbin, 2010) (Marston, 1999) (Kahler, 2008).

3.2 Relazioni con gli Stakeholder

Introduzione

Nel test management, è importante ottimizzare il testing per rilasciare un buon valore di business. Un testing eccessivo può causare ritardi irragionevoli e costi superiori ai benefici, mentre un testing insufficiente può portare a rilasciare agli utenti un prodotto di bassa qualità. L'approccio ottimale si trova tra questi due estremi. È responsabilità del test manager di aiutare gli stakeholder a comprendere questo equilibrio e il valore aggiunto del testing nel raggiungere questo equilibrio, tenendo anche in mente, ad esempio, dei vincoli temporali tipici di un progetto.

3.2.1 Costo della Qualità

I benefici del testing sono compensati dai costi della qualità. Un mezzo per quantificare il costo totale dell'effort e dei difetti legati alla qualità è chiamato costo della qualità. Il costo della qualità consiste nel classificare i costi di progetto e i costi operativi in quattro categorie correlate ai costi dei difetti del prodotto:

- **Costi di prevenzione:** Il costo di tutte le attività che sono state pianificate e sono proattive per prevenire una scarsa qualità (p.e. la qualifica degli sviluppatori per i loro compiti, come la formazione per la creazione di codice manutenibile o sicuro, la review anticipata della base di test e una comunicazione appropriata all'interno del team)
- **Costi di valutazione (appraisal):** Il costo di tutte le attività finalizzate alla rilevazione dei difetti (p.e. l'esecuzione del testing statico e del testing dinamico, e la review dei prodotti di lavoro)
- **Costi di failure interno:** Il costo di tutte le attività reattive (p.e. la correzione dei difetti rilevati durante il testing, il rilascio di workaround)
- **Costi di failure esterno:** Il costo di tutte le attività non a valore aggiunto e reattive (p.e. sia la perdita di entrate, di asset, della salute umana, di vite umane, sia l'ambiente, i costi legali relativi alla correzione dei difetti, il testing, il rilascio e il supporto a causa di un prodotto difettoso rilasciato al cliente ("post release"), la correzione dei difetti sul campo (segnalati dai clienti)).

I costi totali di valutazione e i costi di failure interno sono di solito significativamente inferiori ai costi di failure esterno. Questo rende il testing estremamente prezioso. Per determinare i costi di queste quattro categorie, i test manager possono creare un convincente business case per il testing.

Esistono più approcci che possono essere presi in considerazione per definire il costo della qualità. Il Syllabus ISTQB® ne supporta due. Questo Syllabus si basa sull'approccio di Feigenbaum, mentre ISTQB® Foundation Level Syllabus V.4 presenta l'approccio di Boehm (vedi il Syllabus ISTQB® Foundation Level V.4, Sezione 1.3, Principi del Testing). Questi due approcci sono stati selezionati per raggiungere una comprensione più ampia del costo della qualità. L'approccio di Feigenbaum (Feigenbaum, Nov/Dec 1956) considera la qualità come un processo a livello aziendale e orientato al cliente, mentre l'approccio di Boehm (Boehm, 1979) si focalizza sul compromesso tra il costo della prevenzione e il costo di failure nello sviluppo del software (Hadjicostas, 2004).

3.2.2 Relazione Costi-benefici del Testing

Sebbene la maggior parte delle organizzazioni consideri il testing come prezioso, pochi manager, compresi i test manager, sono in grado di quantificare, descrivere o articolare tale valore. Inoltre, molti test manager, test lead e tester si focalizzano sui dettagli operativi del testing (cioè sugli aspetti specifici del compito del

test o del livello di test), ignorando i problemi tattici e strategici più ampi (di livello più alto) legati al testing, che stanno a cuore ad altri stakeholder, soprattutto ai manager.

Il testing rilascia vantaggi all'organizzazione, al progetto e/o a operations in termini sia quantitativi sia qualitativi:

- **I vantaggi qualitativi** includono una migliore reputazione sulla qualità, rilasci più regolari e prevedibili, aumento della fiducia, protezione da responsabilità legali e riduzione del rischio di perdita di interesse missioni o addirittura di vite umane
- **I benefici quantitativi** includono i difetti rilevati, evitati o corretti prima del rilascio, i difetti che erano noti prima del rilascio (cioè, non corretti ma documentati, magari con workaround), i benefici in termini di costi (Bohm 1981, Böhler 2008), la riduzione del livello di rischio attraverso l'esecuzione dei test, e il rilascio di informazioni sullo stato del progetto, del processo e del prodotto

Un ulteriore vantaggio del testing è che tutti gli stakeholder ottengono informazioni adeguate a prendere decisioni informate, per decidere se la qualità del prodotto sia sufficiente per andare in produzione, con o senza difetti. A volte, andare in produzione con difetti noti è molto meglio che aspettare di andare in produzione fino a quando i difetti non sono stati risolti. I casi in cui un difetto può essere tollerato, dipendono fortemente dalla probabilità che si verifichi il difetto e dalla relativa severità.

Il costo della qualità del testing per difetto viene calcolato come segue:

Risparmio medio per difetto = Media dei costi di failure esterno per difetto - (Costi medi di valutazione per difetto + Media dei costi di failure interno per difetto)

Costo totale della qualità = (Costi di prevenzione dei difetti + (Costi di valutazione * Numero di difetti rilevati prima del rilascio)) + ((Costi di failure interno * Numero di difetti rilevati prima del rilascio) + (Costi di failure esterno * Numero di difetti rilevati dopo il rilascio))

A titolo di esempio, ipotizzate di aver calcolato il seguente costo della qualità per difetto di un prodotto:

- Costi di prevenzione dei difetti: 180€
- Costi medi di valutazione per difetto: 500€
- Media dei costi di failure interno per difetto: 200€
- Media dei costi di failure esterno per difetto: 4.000€

I costi medi di prevenzione dei difetti, i costi di valutazione e i costi di failure interno sono calcolati utilizzando il numero di difetti rilevati prima del rilascio, mentre i costi medi di failure esterno sono calcolati utilizzando il numero di difetti rilevati dopo il rilascio. Con questi valori, possiamo calcolare il risparmio medio per difetto come segue:

Risparmio medio per difetto = 4.000€ - (500€ + 200€) = 3.300€.

La curva di Boehm è una rappresentazione grafica dei costi di correzione dei difetti nel tempo durante il SDLC. Ne consegue che il testing dovrebbe essere eseguito in anticipo nel SDLC per ridurre i costi di correzione dei difetti. La curva di Boehm mostra che i costi di failure interno, o il costo per la correzione di un difetto, aumenta quanto più tardi un difetto viene scoperto nel SDLC. Sulla base di queste informazioni, il test manager dovrebbe cercare di trovare la relazione ottimale dei costi di prevenzione dei difetti rispetto ai costi interni ed esterni.

L'effort del test deve essere basato sul rischio specifico del progetto e del prodotto, e sul rischio che il business è disposto a prendersi. Troppo testing può causare costi più alti rispetto ai benefici derivanti dalla

riduzione del livello di rischio. Se viene testato troppo poco, i difetti che non vengono rilevati possono rappresentare un rischio elevato e generare costi più alti rispetto ai costi dei test non eseguiti. Il testing basato sul rischio (si veda il paragrafo 1.3, *Testing Basato sul Rischio*) supporta la relazione costi-benefici del testing investendo livelli di effort del test in modo proporzionale ai livelli di rischio, e prioritizzando i test in base al loro livello di rischio.

I test manager dovrebbero comprendere quali di questi benefici e costi si applicano all'organizzazione, al progetto e/o a operations, ed essere in grado di comunicare il valore aggiunto del testing in termini di questi benefici e del costo della qualità per difetto.

4 Riferimenti

Standard

- **IEC 61508** (2010) Functional safety of electrical/electronic/programmable electronic safety-related systems - Parts 1 to 7
- **ISO/IEC/IEEE 29119-2** (2021) ISO/IEC/IEEE 29119-2 Software and systems Engineering-Software testing-Part 2 Test processes
- **ISO/IEC/IEEE 29119-3** (2021) ISO/IEC/IEEE 29119-3 Software and systems Engineering-Software testing-Part 3 Test documentation

Documenti ISTQB®

- Syllabus ISTQB® Certified Tester Agile Test Leadership at Scale v2.0 (2023)
- Syllabus ISTQB® Certified Tester Foundation Level V.4 (2023)
- Syllabus ISTQB® Certified Tester Expert Level Test Management v1.0 (2011)
- Syllabus ISTQB® Certified Tester Expert Level Improving the Test Process v1.0.1 (2011)

Libri

- Basili, V., Trendowicz, A., Kowalczyk, M., Heidrich, J., Seaman, C., Münch, J., & Rombach, D. (2014) *Aligning Organizations Through Measurement - The GQM+ Strategies Approach*. Springer International.
- Bath, G., & van Veenendaal, E. (2014). *Improving the Test Process - chapter 6: Process for Improvement*. Rocky Nook.
- Belbin, R. M. (2010). *Management Teams: Why They Succeed or Fail*. London: Routledge.
- Black, R. (2009). *Managing the Testing Process, 3rd Edition*. John Wiley & Sons.
- Boehm, B. (1979). *Software Engineering Economics*. Prentice-Hall.
- Bonebright, D. A. (2010). *40 years of storming: a historical review of Tuckman's model of small group development* (1 Ausg., Bd. 13). Human Resource Development International, 1, 2010, Vol. 13.
- Craig, R., & Jaskiel, S. P. (2002). *Systematic Software Testing*. Artech House.
- Derby, E., & Larsen, D. (2006). *Agile Retrospectives - Making Good Teams Great*. The Pragmatic Bookshelf.
- Erpenbeck, J., & von Rosenstiel, L. (2017). *Handbuch Kompetenzmessung*. Stuttgart: Schäffer-Poeschel.
- Fowler, M. (2010). *Hybrid development processes*. IEEE Software, 27(2), 57-63.
- Herzberg, F., Mausner, B., & Bloch Snyderman, B. (1993). *Motivation to Work*. London: Routledge.

- Kahler, T. (2008). *The Process Therapy Model: The Six Personality Types with Adaptations*. Taibi Kahler Associates, Inc.
- Marston, W. M. (1999). *Emotions Of Normal People*. London: Routledge.
- Sonntag, K., & Schmidt-Rathjens, C. (2005). *Anforderungsanalyse und Kompetenzmodelle*. Wiesbaden: VS Verlag für Sozialwissenschaften.
- Tuckman, B. W. (1965). *Developmental sequence in small groups* (Bd. 63(6)). Psychological Bulletin.
- van Ewijk, A. (2013). *TPI NEXT - Business Driven Test Process Improvement*. Sogeti Nederland B.V.
- van Solingen, R., & Berghout, E. (1999). *The Goal Question Metric Method - A Practical Guide for Quality Improvement of Software Development*. McGraw-Hill.
- van Veenendaal, E. (2012). *The PRISMA Approach: Practical Risk-Based Testing*. UTN Publishers.
- van Veenendaal, E. (2020). *TMMi in the Agile world, version 1.4*. TMMi Foundation.
- van Veenendaal, E., & Cannegieter, J. J. (2011). *The Little TMMi - Objective-Driven Test Process Improvement*. UTN Publishers.

Articoli

- Feigenbaum, Armand V. (Nov/Dec 1956) Harvard Business Review, Vol. 34 Issue 6, p93-101
- Hadjicostas, Evsevios (2004) Total Quality Management and Cost of Quality, Springer https://link.springer.com/chapter/10.1007/978-3-662-09621-5_7

Pagine Web

- www.tmmi.org Test Maturity Model integration (TMMi®); ultima visita 31 gennaio, 2024
- www.tmap.net Test Process Improvement (TPI); ultima visita 31 gennaio, 2024
- www.wikipedia.org/wiki/PDCA Plan-Do-Check-Act; ultima visita 31 gennaio, 2024

I riferimenti precedenti rimandano a informazioni disponibili su Internet e altrove. Anche se tali riferimenti sono stati controllati al momento della pubblicazione di questo Syllabus, ISTQB® non può essere ritenuta responsabile se i riferimenti non sono più disponibili.

5 Appendice A - Obiettivi di Apprendimento/Livello Cognitivo di Conoscenza

Gli obiettivi specifici di apprendimento che si applicano a questo Syllabus sono indicati all'inizio di ogni capitolo. Ogni argomento del Syllabus sarà esaminato in base al suo obiettivo di apprendimento.

Gli obiettivi di apprendimento iniziano con un verbo di azione corrispondente al livello cognitivo di conoscenza, come elencato di seguito.

Livello 1: Ricordare (K1)

Il candidato è in grado di ricordare, riconoscere e richiamare un termine o un concetto.

Verbi di azione: Richiamare, riconoscere.

Esempi
Richiamare i concetti della piramide di test.
Riconoscere gli obiettivi tipici del testing.

Livello 2: Comprendere (K2)

Il candidato è in grado di selezionare le ragioni o le spiegazioni delle affermazioni relative all'argomento ed è in grado di riassumere, confrontare, classificare e fornire esempi per il concetto del testing.

Verbi di azione: Classificare, confrontare, differenziare, distinguere, spiegare, fornire esempi, interpretare, riassumere.

Esempi	Note
Classificare gli strumenti di test in base al loro scopo e alle attività di test che supportano.	
Confrontare i diversi livelli di test.	Può essere utilizzato per cercare somiglianze, differenze o entrambe le cose.
Differenziare il testing dal debugging.	Cerca le differenze tra i concetti.
Distinguere tra rischi di progetto e rischi di prodotto.	Consente di classificare separatamente due (o più) concetti.
Spiegare l'impatto del contesto sul processo di test.	
Fornire esempi del perché il testing è necessario.	
Ricavare la root cause dei difetti da un determinato profilo di failure.	
Riassumere le attività del processo di review del prodotto di lavoro.	

Livello 3: Applicare (K3)

Il candidato è in grado di eseguire una procedura quando si trova di fronte a un compito che gli è familiare, oppure è in grado di selezionare la procedura corretta e applicarla a un determinato contesto.

Verbi di azione: Applicare, implementare, preparare, utilizzare

Esempi	Note
Applicare l'analisi ai valori limite per ricavare i test case da determinati requisiti.	Dovrebbe riferirsi a una procedura/tecnica/processo ecc.
Implementare i metodi di raccolta delle metriche per supportare i requisiti tecnici e di management.	
Preparare test di installabilità per le applicazioni mobile.	
Utilizzare la tracciabilità per monitorare l'avanzamento del test, in modo da poter verificare la completezza e la consistenza con gli obiettivi del test, la strategia di test e il test plan.	Potrebbe essere utilizzato in un LO che richiede che il candidato sia in grado di utilizzare una tecnica o una procedura. Simile ad "applicare".

Livello 4: Analizzare (K4)

Il candidato è in grado di separare le informazioni relative a una procedura o a una tecnica nelle sue parti costituenti per una migliore comprensione, ed è in grado di distinguere tra fatti e inferenze. Un'applicazione tipica è analizzare un documento, un software o una situazione di progetto, e proporre azioni appropriate per risolvere un problema o completare un compito.

Verbi di azione: Analizzare, analizzare in modo critico, delineare, prioritizzare, selezionare

Esempi	Note
Analizzare una determinata situazione di progetto per determinare quali tecniche di test black-box o basate sull'esperienza dovrebbero essere applicate per raggiungere obiettivi specifici.	Esaminabile solo in combinazione con un obiettivo misurabile dell'analisi. Dovrebbe essere della forma "Analizzare xxxx per xxxx" (o similare).
Prioritizzare i test case in una determinata test suite per l'esecuzione, in base ai rischi di prodotto correlati.	
Selezionare i livelli di test e i tipi di test appropriati per verificare un determinato insieme di requisiti.	Necessario quando la selezione richiede un'analisi.

Riferimenti

(Per i livelli cognitivi degli obiettivi di apprendimento)

Anderson, L. W. and Krathwohl, D. R. (eds) (2001) A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives, Allyn & Bacon

Gli obiettivi di apprendimento specifici che si applicano a questo Syllabus sono indicati all'inizio di ogni capitolo.

6 Appendice B - Matrice di Tracciabilità dei Business Outcome rispetto agli Obiettivi di Apprendimento

Questo capitolo elenca la tracciabilità tra i Business Outcome e gli Obiettivi di Apprendimento del Syllabus Advanced Level Test Management.

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM_01	Gestire il testing nei vari progetti di sviluppo software applicando i processi di test management stabiliti per il team di progetto o per l'organizzazione del test		12										
TM_02	Identificare gli stakeholder del test e i modelli del ciclo di vita dello sviluppo software che sono rilevanti in un determinato contesto			4									
TM_03	Organizzare sessioni di identificazione del rischio e di valutazione del rischio all'interno di qualsiasi ciclo di vita dello sviluppo software, e utilizzare i risultati per guidare il testing per raggiungere gli obiettivi del test				6								
TM_04	Definire una strategia di test del progetto consistente con la strategia di test dell'organizzazione e con il contesto di progetto					11							
TM_05	Monitorare e controllare continuamente il testing per raggiungere gli obiettivi del progetto						4						

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM_06	Valutare e fornire report sull'avanzamento del test agli stakeholder di progetto							3					
TM_07	Identificare le competenze necessarie e sviluppare queste competenze all'interno del proprio team								6				
TM_08	Preparare e presentare un business case per il testing in diversi contesti che delinea i costi e i benefici attesi									5			
TM_09	Guidare le attività di test process improvement (miglioramento del processo di test) nei progetti o nei cicli di vita dello sviluppo software di un prodotto, e contribuire alle iniziative di test process improvement dell'organizzazione										5		
TM_10	Pianificare le attività di test includendo l'infrastruttura del test necessaria, e stimare l'effort richiesto per il testing											9	
TM_11	Creare i defect report e un workflow del difetto che siano adeguati al ciclo di vita dello sviluppo software												6
LO unico	Obiettivo di Apprendimento	Livello K											
1	Gestire le Attività di Test												
1.1	Il Processo di Test												
TM-1.1.1	Riassumere la pianificazione dei test	K2	X			X							

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.1.2	Riassumere il monitoraggio dei test e il controllo dei test	K2	X				X						
TM-1.1.3	Riassumere il completamento dei test	K2	X					X					
1.2	Il Contesto del Testing												
TM-1.2.1	Confrontare i motivi per cui differenti stakeholder sono interessati al testing	K2		X		X							
TM-1.2.2	Spiegare perché la conoscenza degli stakeholder è importante nel test management	K2		X		X							
TM-1.2.3	Spiegare il testing in un modello di sviluppo software ibrido	K2		X		X							
TM-1.2.4	Riassumere le attività di test management per i diversi cicli di vita dello sviluppo software	K2	X	X		X							
TM-1.2.5	Confrontare le attività di test management con i diversi livelli di test	K2	X			X							
TM-1.2.6	Confrontare le attività di test management per i diversi tipi di test	K2	X			X							
TM-1.2.7	Analizzare un progetto e determinare le attività di test management che enfatizzano la pianificazione dei test, il monitoraggio dei test e il controllo dei test	K4	X			X							
1.3	Testing Basato sul Rischio												

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.3.1	Spiegare le varie misure adottate nel testing basato sul rischio per poter rispondere ai rischi	K2			X								
TM-1.3.2	Fornire esempi di tecniche differenti che un test manager può utilizzare per identificare i rischi relativi alla qualità del prodotto	K2			X								
TM-1.3.3	Riassumere i fattori che determinano i livelli di rischio relativi alla qualità del prodotto	K2			X								
TM-1.3.4	Selezionare le attività di test appropriate per mitigare i rischi in base al loro livello di rischio in un determinato contesto	K4			X								
TM-1.3.5	Distinguere tra esempi heavyweight e lightweight di tecniche del testing basato sul rischio	K2			X								
TM-1.3.6	Fornire esempi di metriche di successo e di difficoltà associate al testing basato sul rischio	K2			X								
1.4	La Strategia di Test del Progetto												
TM-1.4.1	Spiegare le scelte tipiche per un approccio del test	K2				X							
TM-1.4.2	Analizzare una strategia di test dell'organizzazione e il contesto di progetto per selezionare l'approccio del test appropriato	K4				X							

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.4.3	Utilizzare la metodologia degli obiettivi S.M.A.R.T. per definire obiettivi di test e criteri di uscita misurabili	K3				X							
1.5	Migliorare il Processo di Test												
TM-1.5.1	Spiegare come utilizzare il modello IDEAL per il test process improvement (miglioramento del processo di test) in un determinato progetto	K2									X		
TM-1.5.2	Riassumere l'approccio di miglioramento model-based al test process improvement (miglioramento del processo di test) e comprendere come applicarlo in un contesto di progetto	K2									X		
TM-1.5.3	Riassumere l'approccio di miglioramento analytical-based al test process improvement (miglioramento del processo di test) e comprendere come applicarlo in un contesto di progetto	K2									X		
TM-1.5.4	Implementare una retrospettiva di progetto o di iterazione per valutare i processi di test e scoprire le aree del testing da migliorare	K3									X		
1.6	Strumenti di Test												
TM-1.6.1	Riassumere le best practice per l'introduzione dello strumento	K2										X	

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-1.6.2	Spiegare l'impatto dei diversi aspetti tecnici e di business nella scelta di un tipo di strumento	K2										X	
TM-1.6.3	Analizzare una determinata situazione per creare un piano per la selezione di uno strumento, che copra i rischi, i costi e i benefici	K4										X	
TM-1.6.4	Distinguere le fasi del ciclo di vita dello strumento	K2										X	
TM-1.6.5	Fornire esempi per la raccolta e la valutazione di metriche utilizzando gli strumenti	K2									X	X	
2	Gestire il Prodotto												
2.1	Metriche di Test												
TM-2.1.1	Fornire esempi di metriche per raggiungere gli obiettivi di test	K2					X						
TM-2.1.2	Spiegare come controllare l'avanzamento del test utilizzando le metriche di test	K2					X						
TM-2.1.3	Analizzare i risultati del test per creare test report che consentano agli stakeholder di prendere decisioni	K4					X	X					
2.2	Stima del Test												
TM-2.2.1	Spiegare i fattori che devono essere considerati nella stima del test	K2	X							X		X	

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-2.2.2	Fornire esempi di fattori che possono influenzare le stime del test	K2	X							X		X	
TM-2.2.3	Selezionare una tecnica o un approccio appropriati per la stima del test in un determinato contesto	K4	X							X		X	
2.3	Defect Management												
TM-2.3.1	Implementare un processo di defect management, includendo il flusworkflow del difetto, che può essere utilizzato per monitorare e controllare i difetti	K3											X
TM-2.3.2	Spiegare il processo e i partecipanti necessari per un defect management efficace	K2											X
TM-2.3.3	Spiegare le specifiche del defect management nello sviluppo software Agile	K2	X										X
TM-2.3.4	Spiegare le sfide del defect management nello sviluppo software ibrido	K2	X										X
TM-2.3.5	Utilizzare i dati e le informazioni di classificazione che dovrebbero essere raccolti durante il defect management	K3											X
TM-2.3.6	Spiegare come le statistiche del defect report possono essere utilizzate per sviluppare il process improvement (miglioramento del processo)	K2										X	X
3	Gestire il Team												

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
3.1	Il Team di Test												
TM-3.1.1	Fornire esempi delle competenze tipiche necessarie ai membri del team di test all'interno delle quattro aree di competenza	K2							X				
TM-3.1.2	Analizzare un determinato contesto di progetto per determinare le competenze richieste ai membri del team di test	K4							X				
TM-3.1.3	Spiegare le tecniche tipiche per la valutazione delle competenze dei membri del team di test	K2							X				
TM-3.1.4	Distinguere tra gli approcci tipici per lo sviluppo delle competenze dei membri del team di test	K2							X				
TM-3.1.5	Spiegare le competenze richieste per gestire un team di test	K2							X				
TM-3.1.6	Fornire esempi di fattori motivanti e di fattori di igiene per i membri del team di test	K2							X				
3.2	Relazione con gli Stakeholder												
TM-3.2.1	Fornire esempi per ognuna delle quattro categorie che determinano il costo della qualità	K2								X			

Business: Advanced Level Test Management			BO1	BO2	BO3	BO4	BO5	BO6	BO7	BO8	BO9	BO10	BO11
TM-3.2.2	Applicare un calcolo costi-benefici per stimare il valore aggiunto del testing per gli stakeholder	K3						X		X			

7 Appendice C - Release Note

Il Syllabus ISTQB® Advanced Level Test Management v3.0 è un aggiornamento importante (major release) basato sul Syllabus Advanced Level Test Manager 2012 (CTAL TM 2012). Per questo motivo, non sono presenti release note dettagliate per capitolo e per paragrafo. Tuttavia, di seguito viene fornito un riepilogo delle principali modifiche.

In questa versione tutti gli Obiettivi di Apprendimento (LO) sono stati modificati per renderli atomici, e per creare una tracciabilità uno-a-uno tra LO e paragrafi del Syllabus, in modo da non avere un contenuto senza avere anche un LO. L'obiettivo è rendere questa versione più facile da leggere, comprendere, imparare e tradurre, focalizzandosi sull'aumento dell'utilità pratica e sull'equilibrio tra conoscenze e competenze.

Questa major release ha apportato le seguenti modifiche:

- Riduzione della dimensione complessiva del Syllabus. Il Syllabus non è un libro di testo, ma un documento che serve a delineare gli elementi di base di un corso avanzato sul testing del software, includendo quali argomenti dovrebbero essere coperti e a quale livello. In particolare:
 - Nella maggior parte dei casi gli esempi sono esclusi dal testo. È compito di un Training Provider fornire gli esempi e gli esercizi durante la formazione
 - È stata seguita la "Checklist per la stesura del Syllabus", che suggerisce la dimensione massima del testo per LO a ciascun livello K (K2 = massimo 1.500 caratteri non vuoti, K3 = massimo 2.500 caratteri non vuoti, K4 = massimo 3.000 caratteri non vuoti, +/- 20%)
- Riduzione del numero di LO rispetto al Syllabus CTAL TM 2012
 - 36 LO Livello K2 rispetto ai 39 LO in CTAL TM 2012
 - 5 LO Livello K3 rispetto ai 12 LO in CTAL TM 2012
 - 7 LO Livello K4 rispetto ai 10 LO in CTAL TM 2012
- La struttura completa del Syllabus è stata rivista
- L'allineamento con il Syllabus ISTQB® Foundation Level V.4 è completo
- Principali modifiche al CTAL TM 2012 ex Capitolo 1 (Processo di Test)
 - Limitato al Capitolo "*Gestire le Attività di Test*" (Pianificazione dei Test, Monitoraggio e Controllo dei Test e Completamento dei Test)
 - Integrato come paragrafo nel nuovo capitolo "*Gestire le Attività di Test*"
- Nuovo capitolo **Gestire le Attività di Test**
 - Paragrafo 1.1 - *Il Processo di Test*: vedi sopra
 - Paragrafo 1.2 - *Il Contesto del Testing*: Ampliata per coprire i modelli dello sviluppo software non sequenziali
 - Paragrafo 1.3 - *Testing Basato sul Rischio*: Completamente riscritta per renderla più applicabile a livello di progetto

- Paragrafo 1.4 - *La Strategia di Test del Progetto*: Poiché il test plan è già definito nel Syllabus ISTQB® Foundation Level V.4, il focus è sulla selezione dell'approccio del test adeguato e sulla definizione di obiettivi del test misurabili
- Paragrafo 1.5 - *Migliorare il Processo di Test*: Integrare questo aspetto in *Gestire le Attività di Test*, mostrare come applicarlo in un contesto di progetto e implementarlo utilizzando le retrospettive all'interno di un'iterazione o di un progetto
- Paragrafo 1.6 - *Strumenti di Test*: L'introduzione degli strumenti è stata spostata dal Syllabus ISTQB® Foundation Level Syllabus V.3.1 (non è presente nel Syllabus ISTQB® Foundation Level V.4)
- Nuovo capitolo **Gestire il Prodotto**
 - Paragrafo 2.1 - *Metriche di Test*: Precedenti paragrafi che definiscono le metriche e l'uso delle metriche, Metriche di Test
 - Paragrafo 2.2 - *Stima del Test*: Il Syllabus ISTQB® Foundation Level V.4 copre già il calcolo della stima del test. Ampliata per selezionare a livello K4 adeguate tecniche di stima del test e l'uso delle stime del test attraverso i modelli del SDLC
 - Paragrafo 2.3 - *Defect Management*: Allineata alle ultime edizioni degli standard e ampliata all'utilizzo nello sviluppo software Agile e nello sviluppo software ibrido
- Nuovo capitolo **Gestire il Team**
 - Paragrafo 3.1 - *Il Team di Test*: Gli argomenti principali sono gli stessi del Syllabus CTAL TM 2012. Identificare le competenze individuali e comporre i team di test
 - Paragrafo 3.2 - *Relazioni con gli stakeholder*: Si tratta del precedente paragrafo del Syllabus CTAL TM 2012 "Valore di Business del Testing"
- Principali modifiche e paragrafi/capitoli eliminati rispetto al CTAL TM Syllabus 2012
 - Eliminato il paragrafo sul Testing Distribuito, Outsourced e Insourced
 - Eliminato il paragrafo sul Gestire l'Applicazione degli Standard Industriali
 - Eliminato il capitolo sulle Review
 - Eliminati i sotto-paragrafi sul Miglioramento del Processo di Test con CTP e STEP
 - Sot Eliminati i sotto-paragrafi sull'Analisi dei Test, sulla Progettazione dei Test, sull'Implementazione dei Test e sull'Esecuzione dei Test

8 Appendice D - Parole Chiave Specifiche del Dominio

Termine	Definizione
Goal-Question-Metric (GQM)	Un approccio alla misurazione del software che utilizza un modello a tre livelli che consiste di un livello concettuale (Goal), un livello operativo (Question) e un livello quantitativo (Metric).
IDEAL	Un modello di miglioramento organizzativo che serve come roadmap per avviare, pianificare e implementare azioni di miglioramento.
indicatore	Una misura che fornisce una stima o una valutazione di attributi specifici derivati da un modello, rispetto a esigenze informative definite.
misura	Il numero o la categoria assegnati a un attributo di un'entità attraverso una misurazione.
metrica	Una scala di misurazione e il metodo usato per la misurazione.
Planning poker	Una tecnica di stima basata sul consenso, principalmente utilizzata per stimare l'effort o la dimensione relativa delle user story nello sviluppo software Agile. È una variante del metodo Wideband Delphi, che utilizza un mazzo di carte con valori che rappresentano le unità di stima del team.
Three-point estimation	Una tecnica expert-based, dove gli esperti eseguono tre stime: la stima più ottimistica (a), la stima più probabile (m) e la stima più pessimistica (b). La stima finale (E) è la loro media aritmetica pesata.
Wideband Delphi	Una tecnica di stima del test expert-based (basata su esperti) che ha lo scopo di ottenere una stima accurata utilizzando la saggezza collettiva dei membri del team.

9 Appendice E – Marchi Registrati

CMMI® è un marchio registrato presso U.S. Patent and Trademark Office dalla Carnegie Mellon University.

ISTQB® è un marchio registrato di International Software Testing Qualifications Board.

TMMi® è un marchio registrato di TMMi Foundation.

TPI-Next® è un marchio registrato di Sogeti, The Netherlands.

10 Indice

Tutti i termini sono definiti nel Glossario ISTQB® (<http://glossary.istqb.org/>).

analisi del rischio; 15; 27; 29; 31; 32
anomalia; 47; 56; 60; 61
avanzamento del test; 10; 12; 13; 17; 18; 20;
30; 40; 47; 49; 50; 51; 60; 61; 76; 79
ciclo di vita dello sviluppo software; 10; 15;
19; 20; 78; 80; 81
comitato pdi defect management; 58
completamento dei test; 12; 15; 17; 19; 40;
48; 49; 50; 65; 80; 86
contenimento di fase; 56; 62
controllo dei test; 12; 15; 17; 18; 26; 30; 48;
49; 65; 79; 80; 86
costo della qualità; 56; 63; 70; 71; 72; 85
defect report; 10; 19; 47; 56; 57; 58; 59; 60;
61; 62; 79; 84
difetto; 19; 47; 51; 56; 57; 58; 59; 60; 61; 63;
71; 72; 79; 84
failure; 29; 31; 46; 47; 56; 57; 58; 61; 63; 75
failure esterno; 63; 70; 71
failure interno; 30; 63; 70; 71
workflow del difetto; 10; 12; 47; 56; 59; 79; 84
identificazione del rischio; 10; 11; 27; 28; 32;
78
impatto del rischio; 15; 28; 29; 31; 32
livello di rischio; 15; 27; 28; 29; 30; 32; 71; 72;
81
livello di test; 14; 15; 17; 19; 23; 30; 33; 34;
35; 50; 51; 52; 54; 65; 66; 71; 75; 76; 80
metodologia degli obiettivi S.M.A.R.T.; 15; 16;
35; 81
metrica; 12; 15; 16; 23; 31; 32; 37; 39; 40; 45;
46; 47; 48; 49; 50; 51; 61; 62; 76; 81; 83;
87; 88 metriche di progetto; 49
mitigazione del rischio; 15; 25; 27; 29
modello di sviluppo software ibrido; 13; 14;
15; 22; 35; 47; 53; 59; 80; 84; 87
modello di sviluppo incrementale; 14; 15; 37
modello di sviluppo iterativo; 14; 15; 23; 53
modello di sviluppo sequenziale; 13; 15; 23;
33; 37; 41; 55; 57; 59; 60; 66
monitoraggio dei test; 12; 15; 17; 18; 19; 26;
27; 30; 48; 49; 65; 79; 80
monitoraggio del rischio; 15; 27
obiettivo del test; 10; 12; 13; 14; 15; 16; 17;
18; 19; 20; 24; 35; 36; 47; 48; 49; 50; 52;
66; 76; 78
pianificazione dei test; 12; 15; 17; 18; 25; 27;
28; 30; 31; 33; 38; 48; 52; 65; 79; 80
planning poker; 47; 54; 55; 88
prevenzione dei difetti; 56; 62; 63
priorità; 26; 30; 31; 32; 34; 46; 51; 58; 61
probabilità del rischio; 15; 18; 28; 29; 31; 32
retrospettiva; 12; 15; 16; 19; 28; 31; 37; 39;
40; 41; 62; 67; 82
rischio di prodotto; 15; 18; 27; 28; 49; 50; 51;
65; 75; 76
rischio di qualità; 15; 27; 28; 29; 31; 32; 50
risk management; 15; 21; 25; 27; 28; 31
risultato falso-negativo; 55
root cause; 31; 32; 39; 40; 51; 61; 62; 75
severità; 31; 46; 51; 53; 61; 71

stima del test; 13; 47; 52; 53; 54; 83

strategia di test; 10; 12; 13; 14; 15; 16; 17; 18;
20; 21; 23; 24; 33; 34; 35; 39; 48; 59; 65;
66; 76; 78; 81

est Maturity Model integration (TMMi); 15; 38;
39

test plan; 14; 15; 17; 18; 19; 20; 21; 23; 25;
33; 35; 60; 76

test process improvement; 10; 12; 15; 16; 17;
37; 38; 39; 40; 49

testing basato sul rischio; 12; 13; 15; 16; 18;
27; 28; 30; 31; 32; 34; 62; 72; 80; 81

testing basato sull'esperienza; 15; 34

testing funzionale; 15; 24

testing non-funzionale; 15; 24; 25

tipo di test; 14; 15; 17; 24; 30; 33; 34; 35; 45;
54; 65; 76; 80

TPI Next; 15; 38; 39

valutazione (appraisal); 63, 70; 71

valutazione del rischio; 10; 15; 24; 27; 28; 31;
78