

Software Testing

Processo di Testing
Livelli di Testing
Deliverables del processo di testing

1

1

Pianificazione del Testing

- ❑ Un'attenta pianificazione è necessaria per ottenere i risultati migliori dai processi di ispezione e testing
- ❑ La pianificazione dovrebbe partire presto nel processo di sviluppo
- ❑ Il piano dovrebbe individuare un bilanciamento tra verifica statica e testing
- ❑ La pianificazione del test è basata sulla definizione di standard per il processo di testing e i suoi deliverable

2

2

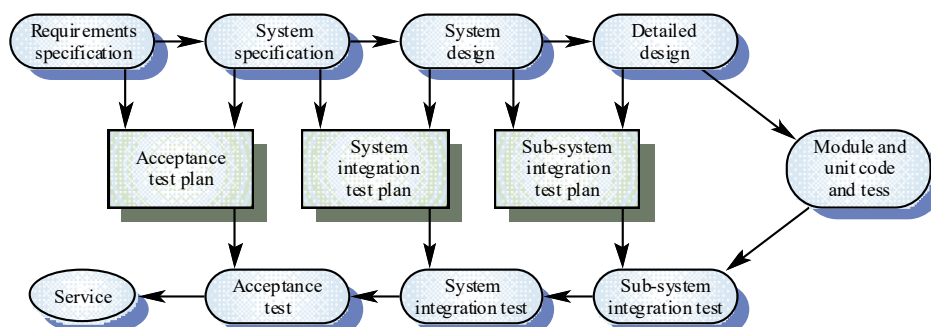
Testing e Processo di Sviluppo

- ❑ il test non è una fase
 - ✓ attività di analisi e test sono (e devono essere) presenti per tutta la durata del ciclo di vita (controllo di qualità)
 - ✓ il processo deve essere pianificato per garantire visibilità il prima possibile e continuamente
- ❑ i test sono parte del prodotto finale
 - ✓ devono essere sviluppati con codice e documentazione
 - ✓ devono essere salvati per manutenzione
- ❑ progettare per il test
 - ✓ la struttura del processo e del prodotto possono essere adattati per facilitare e migliorare la misura di qualità

3

3

Il modello di sviluppo a V



4

4

Attività di Testing e Processo (1)

identificazione dei requisiti	specificazione dei requisiti	progetto di alto livello	progetto di dettaglio	◆◆◆
<ul style="list-style-type: none"> • identificazione qualità • pianificazione test di accettazione 	<ul style="list-style-type: none"> • convalida specifiche • pianificazione test di sistema • creazione test funzionali 	<ul style="list-style-type: none"> • ispezione progetto di alto livello • pianificazione test di unità e di integrazione • analisi automatica del progetto di alto livello 	<ul style="list-style-type: none"> • ispezione progetto • generazione oracoli • generazione casi di test black-box • analisi automatica del progetto 	

5

5

Attività di Testing e Processo (2)

◆◆◆	codifica e test di unità	integrazione test di sistema	accettazione e rilascio	operatività e manutenzione
	<ul style="list-style-type: none"> • ispezione codice • creazione scaffolding • esecuzione test di unità • analisi automatica del codice • analisi di copertura 	<ul style="list-style-type: none"> • esecuzione test di integrazione • esecuzione test di sistema 	<ul style="list-style-type: none"> • esecuzione test di accettazione • consegna test di regressione 	<ul style="list-style-type: none"> • esecuzione test di regressione • revisione test di regressione

6

6

Chi Deve Fare Testing

Lo sviluppatore



Ha sviluppato il sistema quindi i suoi test sono inquinati da aspetti psico-cognitivi: perché deve distruggere il suo sistema?

Obiettivo: la qualità interna

Un team esterno



Deve comprendere il sistema e fa tutti gli sforzi per poterlo distruggere

Obiettivo: la qualità esterna

7

7

Livelli di Testing

□ Produttore:

- ✓ unit testing
(testing di unità)
- ✓ integration testing
(testing di integrazione)
- ✓ system testing
(testing di sistema)

□ Cooperazione Produttore/Cliente

- ✓ alpha testing
- ✓ beta testing

□ Cliente

- ✓ acceptance testing
(testing di accettazione)

8

8

Testing di Unità

- ❑ Il testing è applicato isolatamente ad una unità (modulo) di un sistema software
- ❑ Obiettivo fondamentale è quello di rilevare errori (logica e dati) nel modulo
- ❑ prassi diffusa è che venga realizzato direttamente dal programmatore che ha sviluppato il modulo l'unità sottoposta a test

unità: elemento definito nel progetto di un sistema software e testabile separatamente

unità e modulo sono spesso usati come sinonimi

9

9

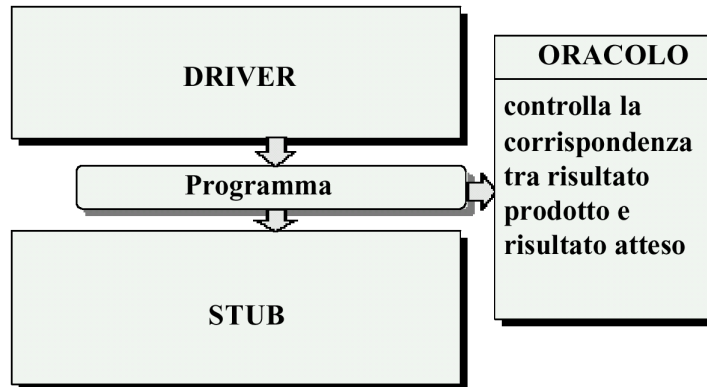
Il problema dello scaffolding

- ❑ Creare l'ambiente per l'esecuzione dei test
 - ✓ Lo scaffolding è estremamente importante per il test di unità e integrazione
 - ✓ Può richiedere un grosso sforzo programmatico
 - ✓ Uno scaffolding buono è un passo importante per test di regressione efficiente
 - ✓ La generazione di scaffolding può essere parzialmente automatizzato a partire dalle specifiche ...

10

10

Creare scaffolding



La necessità di moduli stub e di moduli driver dipende dalla posizione del modulo nell'architettura del sistema ...

11

11

Driver e Stub

- ❑ Driver (Modulo Guida)
 - ✓ Deve simulare l'ambiente chiamante
 - ✓ Deve occuparsi dell'inizializzazione dell'ambiente non locale del modulo in esame
- ❑ Stub (Modulo Fittizio)
 - ✓ Ha la stessa interfaccia del modulo simulato, ma è più semplice

12

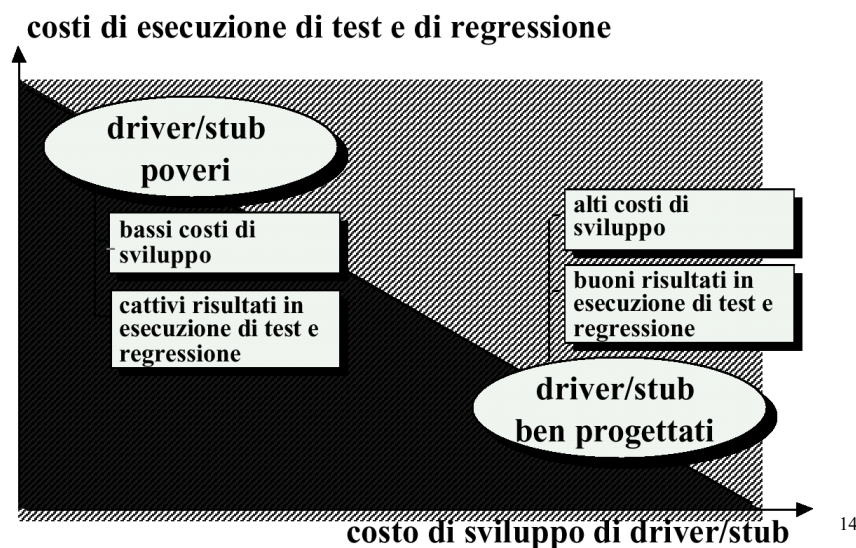
12

Generare Driver e Stub

- ❑ Interattivi
 - ✓ Richiesto l'intervento umano
 - ✓ Troppo oneroso ... e l'utente potrebbe introdurre errori ...
- ❑ Automatici
 - ✓ Driver: inizializzazioni costanti dell'ambiente non locale
 - ✓ Stub: Calcola valore approssimato, restituzione di valori costanti
 - » A volte non basta ... causa di malfunzionamenti
- ❑ Realizzazione di prototipi
 - ✓ realizzazione rapida e con costi ridotti, a scapito dell'efficienza
- ❑ Costruiti a partire da specifiche
 - ✓ stesse metodologie di sviluppo del software da rilasciare ...¹³

13

Costruzione di Driver e Stub: Tradeoff



14

Software OO e Unit Testing

- ❑ I livelli tradizionali mal si adattano al caso di linguaggi OO
 - ✓ da cosa è rappresentata l'unità?
 - ✓ cosa è un modulo in un sistema OO?
 - ✓ esistono diverse scuole di pensiero
- ❑ Una possibile suddivisione:
 - ✓ *Basic unit testing*: test di una singola operazione di una classe (intra-method testing)
 - ✓ *Unit testing*: test di una classe nella sua globalità (intra-class testing)

15

15

Stato e information hiding

- ❑ Linguaggi procedurali standard
 - ✓ componente base: procedura
 - ✓ metodo di test: test della procedura basato su input/output
- ❑ Linguaggi object-oriented
 - ✓ componente base: Classe = struttura dati + insieme di operazioni
 - ✓ oggetti sono istanze di classi
 - ✓ la correttezza non è legata solo all'output, ma anche allo stato, definito dalla struttura dati
 - ✓ lo stato "privato" può essere osservato solo utilizzando metodi *pubblici* della classe (e quindi affidandosi a codice sotto test)

16

16

Scaffolding per software OO

- ❑ L'infrastruttura deve
 - ✓ settare opportunamente lo stato per poter eseguire i test (driver)
 - ✓ esaminare lo stato per poter stabilire l'esito dei test (oracoli)
 - ✓ ma lo stato è "privato" ...
- ❑ Approcci intrusivi:
 - ✓ modificare il codice sorgente
 - » Aggiungere un metodo testdriver alla classe ...
 - ✓ usare costrutti del linguaggio (esempio: costrutto friend)

17

17

Testing di Integrazione

- ❑ E' il testing applicato ad un aggregato di due o più unità di un sistema software
- ❑ Obiettivo fondamentale è quello di rilevare degli errori nella integrazione fra le unità e nelle funzioni che l'aggregato deve assolvere
- ❑ Non è compito dei programmatori che hanno prodotto le unità componenti
- ❑ Talvolta il termine test di integrazione viene riferito anche alla integrazione fra componenti hardware e software

18

18

Errori di Integrazione

❑ ***Interpretation Error***

- ✓ Funzionalità/comportamento del modulo diversi da quelli attesi dall'utente del modulo

❑ ***Miscoded Call Error***

- ✓ Istruzione di chiamata (messaggio) inserita nel posto sbagliato (o non inserita dove richiesta)

❑ ***Interface Error***

- ✓ Violazione dell'interfaccia standard tra due moduli (cattiva specifica/implementazione di interfacce)

❑ ***Uso di drivers/stubs poveri durante il test di unità***

19

19

Testing di Integrazione: modalità

❑ ***Big bang test*** (test non incrementale)

- ✓ testare prima tutti i moduli, quindi integrare i moduli testati e verificare l'intero sistema

❑ ***Testing incrementale***

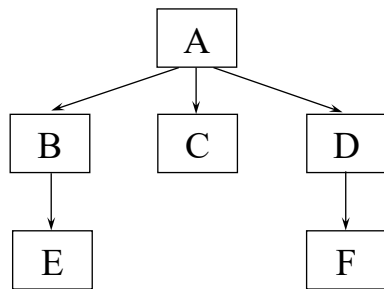
- ✓ Integrazione dei moduli man mano che vengono prodotti e testati ... molti vantaggi:
 - » meno moduli fittizi e moduli guida
 - » individuazione ed eliminazione di anomalie sulle interfacce durante lo sviluppo del sistema
 - » anomalie più facilmente localizzabili e rimovibili
 - » ciascun modulo è esercitato più a lungo

20

20

Testing di Integrazione per software tradizionale (procedurale)

- ❑ Le unità da integrare sono ricavate in base a criteri funzionali ricavabili dall'architettura del sistema



- ❑ Strategie incrementali
 - ✓ *top-down, bottom-up, sandwich* (combina top-down e bottom-up)
 - ✓ altre combinazioni di top-down, bottom-up e sandwich: *modified top-down, modified sandwich, build testing, mixed bag ...*

21

21

Testing incrementale top-down

- ❑ Sviluppare, testare ed integrare prima i moduli più in alto nella gerarchia delle chiamate (moduli di coordinamento)
 - ✓ Possibili depth-first (in profondità) or breadth-first (in ampiezza)
- ❑ Non richiede moduli driver ... ma richiede moduli stub piuttosto complessi
- ❑ Postpone il testing di moduli di ingresso/uscita e di moduli di elaborazione

22

22

Testing incrementale bottom-up

- ❑ Sviluppo e test prima dei moduli periferici della gerarchia delle chiamate
- ❑ Necessità di moduli driver, non richiesti moduli stub
- ❑ Subito il test dei moduli terminali, tra cui moduli di elaborazione ed interfaccia complessi
- ❑ Ritardato il test dei moduli di controllo

23

23

Testing di integrazione per software OO

- ❑ Big bang: in generale poco adatto
- ❑ Top-down e bottom-up: cambia il tipo di dipendenze tra “moduli”
 - ✓ Dipendenze: uso di classi ed ereditarietà
 - » Se A usa B allora A dipende da B
 - » Se A eredita da B allora A dipende da B
 - ✓ Preferibile una strategia bottom-up (testare prima le classi indipendenti)
 - » Stub troppo difficili da costruire

24

24

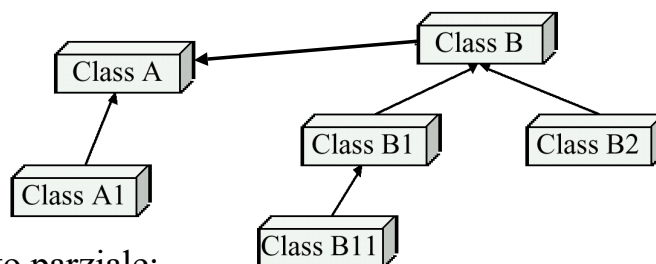
Grafo delle dipendenze

- ❑ Le dipendenze tra le classi possono essere espresse su un grafo delle dipendenze
- ❑ Se il grafo è aciclico esiste un ordinamento parziale sui suoi elementi:
 - ✓ Possibile definire un ordinamento totale topologico
 - » Privilegiare le dipendenze di specializzazione
 - ✓ Ordine d'integrazione definito in base a tale ordinamento
- ❑ Se esistono dipendenze cicliche tra le classi è impossibile definire un ordinamento parziale, ma ...
 - ✓ Ogni grafo orientato ciclico è riducibile a un grafo aciclico collassando i sottografi massimi fortemente connessi

25

25

Grafo delle dipendenze aciclico

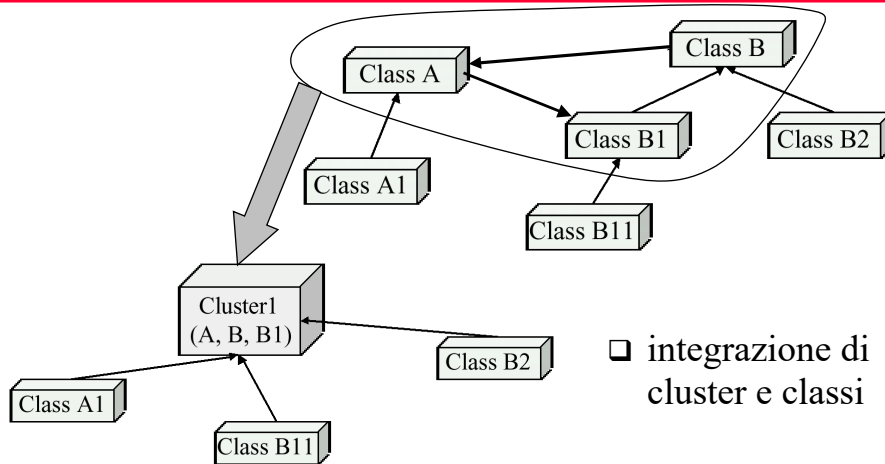


- ❑ ordinamento parziale:
 - ✓ $A < A1$
 - ✓ $A < B$
 - ✓ $B < B1 < B11$
 - ✓ $B < B2$
- ❑ Possibile ordinamento totale: $A < A1 < B < B2 < B1 < B11$

26

26

Grafo delle dipendenze ciclico



impossibile applicare la tecnica sul singolo cluster ... 27

27

Problemi di Integrazione per software OO

- Una volta definito l'ordine di integrazione si aggiungono le classi incrementalmente esercitandone le interazioni
- Possibili problemi di integrazione:
 - ✓ Ereditarietà implica problemi in caso di modifiche di superclassi ...
 - ✓ Polimorfismo comporta problemi legati al binding dinamico ...

28

28

Testing di Sistema

- ❑ E' il testing applicato sul sistema software completo ed integrato
- ❑ L'obiettivo è quello di valutare l'adesione del sistema ai requisiti specificati
- ❑ Va eseguito dal team addetto al testing (esterno al gruppo di sviluppo)

29

29

Testing di Sistema

- ❑ I requisiti di sistema non sono solo le funzionalità esterne
- ❑ Fondamentali possono essere i requisiti di qualità, di prestazioni, stabiliti ad esempio sulla base di un modello (o profilo) di qualità del prodotto opportunamente istanziato
- ❑ Prestazioni, Manutenibilità, Usabilità ...

30

30

Testing di Sistema

- ❑ Funzionale: funzionalità come viste dall'operatore
- ❑ Configurazione: tutti i comandi, gli automatismi per scambiare/cambiare le relazioni fisiche logiche dei componenti HW
- ❑ Recovery: capacità di reazione del sistema a cadute
- ❑ Stress: affidabilità in condizione di carico limite
- ❑ Sicurezza: invulnerabilità del sistema rispetto ad accessi non autorizzati

31

31

Testing di Accettazione

- ❑ Testing effettuato sull'intero sistema sulla base di un piano e di procedure approvate dal cliente
- ❑ L'obiettivo è quello di mettere il cliente, l'utente o altri a ciò preposti (collaudatori o enti ad hoc) in condizioni di decidere se accettare il prodotto
- ❑ E' a carico del committente
- ❑ Segna il passaggio del sistema dal produttore all'ambiente operativo
- ❑ Può talvolta essere più una demo che un test

32

32

Alpha e Beta Testing

❑ Alpha Testing

- ✓ Uso del sistema da parte di utenti reali ma nell'ambiente di produzione e prima della immissione sul mercato
- ✓ Talvolta riferito per il testing da parte di un cliente (o gruppo di clienti) privilegiato(i)

❑ Beta Testing

- ✓ Installazione ed uso del sistema in ambiente reale e prima della immissione sul mercato
- ✓ Strategia adottata da produttori di packages per mercato di massa
- ✓ Talvolta il beta testing è preceduto da un alpha testing, o un beta testing da parte di un gruppo più ristretto di utenti
- ✓ Problemi di confidenzialità

33

33

Deliverables di un processo di testing

- ❑ Il testing è un processo
- ❑ Esigenza di definire modelli di riferimento a partire dai quali instanziare tali processi
- ❑ Un modo per fissare riferimenti comuni per tali processi: definire i deliverables che devono essere prodotti nelle fasi e con le attività che lo caratterizzano
- ❑ I deliverables del processo:
 - ✓ *documenti organizzativi e di management*
 - ✓ *documenti di specifica ed esecuzione*

34

34

Standard ISO/IEC/IEEE 29119-2022

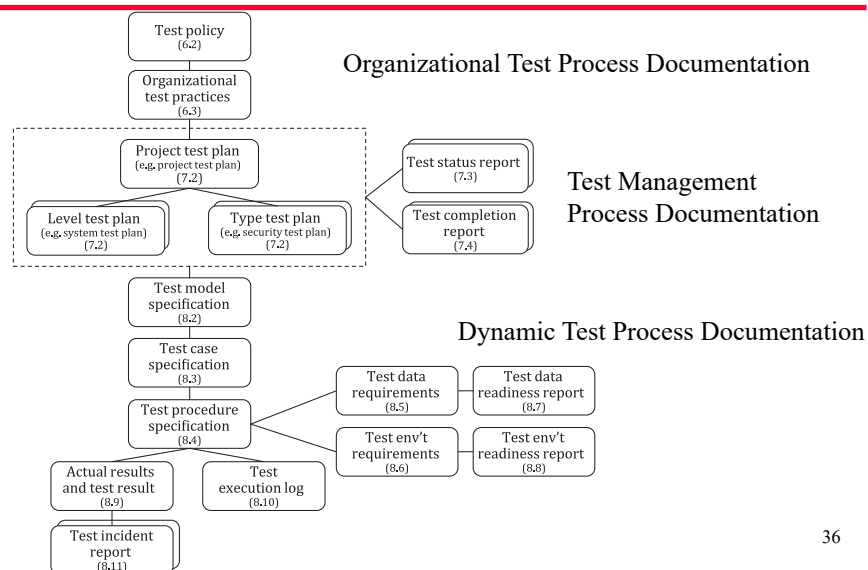
- ❑ Strutturato in tre parti:
 - ✓ General Concepts (Part 1)
 - ✓ Test Processes (Part 2)
 - ✓ **Test Documentation** (Part 3)

- ❑ Lo standard sostituisce il precedente standard IEEE 829 del 2008 (Software and System Test Documentation)

35

35

Documenti di testing: Overview



36

36

Organizational Test Process Documentation

- ❑ Descrivono le informazioni sul testing a livello organizzativo e indipendentemente dal progetto e includono
 - ✓ **Test Policy:** Definisce obiettivi e i principi dell'applicazione del testing nell'organizzazione
 - ✓ **Organizational Test Practices:** fornisce linee guida di come il testing dovrebbe essere condotto

37

37

Test Policy

- ❑ Definisce obiettivi e i principi dell'applicazione del testing nell'organizzazione
- ❑ Definisce cosa dovrebbe essere effettuato mediante il testing, es. i processi, la struttura organizzativa e altri aspetti, senza i dettagli di come il testing dovrebbe essere condotto
- ❑ Fornisce un framework per stabilire, rivedere, e migliorare i processi dell'organizzazione

38

38

Organizational Test Practices

- ❑ Fornisce linee guida di come condurre il testing e ottenere gli obiettivi stabiliti dalla test policy
- ❑ Le Test practice sono definite per ambiti specifici, come un particolare livello di testing (unità, integrazione, sistema) e/o per uno specifico tipo di testing

39

39

Esempio di Organizational Test Practices

Organizational test practices

- Organization-level test practice statements
 - Risk management approach
 - Test selection and prioritization
 - Test documentation and reporting
 - Test automation and tools
 - Configuration management of test work products
 - Incident management
 - Test levels
 - Test types
 - Rules/guidelines for deviation from organizational test practices

Incident: anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system

Test level 1 (e.g. component test)

Entry and exit criteria
Test completion criteria
Test documentation and reporting
Degree of independence
Test design techniques
Test data
Test environment
Metrics to be collected
Retesting and regression testing

Test level 2 (e.g. integration test)

Entry and exit criteria
Test completion criteria
Test documentation and reporting
Degree of independence
Test design techniques
Test data
Test environment
Metrics to be collected
Retesting and regression testing

Test type 1 (e.g. security test)

Entry and exit criteria
Test completion criteria
Test documentation and reporting
Degree of independence
Test design techniques
Test data
Test environment
Metrics to be collected
Retesting and regression testing

40

40

Test Management Process Documentation

- ❑ Comprende i seguenti tipi di documenti:
 - ✓ **Test plan:** documento di pianificazione e gestione del testing
 - ✓ **Test status report:** fornisce informazioni circa lo stato del testing che è eseguito in un particolare periodo di reporting
 - ✓ **Test completion report:** fornisce un summary del testing effettuato. Può essere a livello di progetto o di programma

41

41

Test Plan (1)

- ❑ Descrive le decisioni prese durante la fase iniziale di pianificazione e evolve con le ri-pianificazioni.
- ❑ Descrive l'organizzazione complessiva delle attività di testing, lo schedule, le risorse e le responsabilità, i tool, le strategie, le tecniche e i metodi necessari per eseguire il testing, i rischi, le assunzioni, gli standard e le metriche da raccogliere, le condizioni sotto le quali eseguire regression testing, criteri di sospensione e ripresa del testing.
- ❑ Se un'organizzazione non ha definito delle organizational test practices, allora alcuni di questi aspetti possono essere inclusi nel test plan

42

42

Test Plan (2)

- ❑ Alcuni progetti hanno un solo test plan, progetti più grossi possono avere multiple test plan.
- ❑ I piani di test possono essere applicati a progetti multipli (a livello di programma), or a singoli progetto (es. project test plan/master test plan), or a specifici livelli o tipi di testing (es. system test plan, integration software test plan, subsystem test plan, subcontractor software test plan, unit software test plan, or performance test plan), o a specifiche iterazioni del testing

43

43

Test Completion Report

- ❑ Fornisce un summary del testing effettuato.
- ❑ Può essere a livello di progetto o di programma, oppure per un particolare livello o tipo di test (dipende dal modo in cui sono strutturati i piani di test)
- ❑ Descrive le deviazioni rispetto alle attività pianificate, una valutazione complessiva, eventuali fattori che impedivano i progressi e come sono stati rimossi, misure del processo di testing, rischi residui e lezioni apprese.

44

44

Dynamic Test Process Documentation

❑ Comprende i seguenti tipi di documenti:

- ✓ Test Specification
 - » Test Model Specification
 - » Test Case Specification
 - » Test Procedure Specification
- ✓ Test Data Requirements
- ✓ Test Environment Requirements
- ✓ Test Data Readiness Report
- ✓ Test Environment Readiness Report
- ✓ Test Execution Documentation
 - » Actual results
 - » Test Results
 - » Test Execution Log
- ✓ Incident Report

45

45

Test Specification: Test Model Specification

- ❑ Un test model è una rappresentazione di un aspetto del test item, focalizzato sugli attributi che definiscono la test coverage richiesta.
 - ✓ Test item: a work product that is an object of testing.
 - » Can be either a complete system or a part of a larger system, including hardware, software, and related documentation.
 - » Can be either executable (e.g. binary code, some models) or not executable (e.g. a documented specification).
 - ✓ Esempi di test model: classi di equivalenza (e quindi anche le scelte in cui si suddividono le categorie), tabelle di decisione, state transition diagram, source code, control flow graph, etc.
- ❑ La specifica di un test model deve includere un identificatore univoco, obiettivi, priorità, estratto della test strategy, tracciabilità sui requisiti

46

46

Test Specification: Test Case Specification (1)

- ❑ Identifica gli elementi di test coverage e i corrispondenti test case derivati applicando una o più tecniche al test model.
- ❑ Test Coverage items
 - ✓ Riassume i test coverage items identificati dal test model e derivati applicando una tecnica al test model.
 - » Ad esempio, in weak equivalence class testing i test coverage item sono le singole classi di equivalenza, in strong equivalence class testing (o in category partition), i test coverage item saranno le combinazioni di classi di equivalenza (o le combinazioni di scelte).
 - » Nel white box testing, i test coverage items sono i singoli statement, o i branch, o i path del control flow graph, a seconda del criterio di copertura adottato ...
 - ✓ Nella specifica di un test coverage item va incluso un identificatore univoco, la descrizione, la priorità, la tracciabilità sul test model

47

Test Specification: Test Case Specification (2)

- ❑ Test Cases
 - ✓ Questa parte della test case specification definisce i casi di test derivati dai test coverage items
 - » Un test case specifica come uno o più test coverage item sono esercitati (ad esempio, lo stesso caso di test può esercitare più classi di equivalenza in weak equivalence class testing) o più branch nel criterio di branch coverage
 - » Il numero di casi di test derivati dai test coverage item dipende dal test coverage completion criterion definito nel test plan (ad esempio, se solo l'80% di test coverage è richiesto, allora non tutti i test coverage item devono essere associati a casi di test)
 - ✓ Nella specifica di un caso di test va incluso un identificatore univoco, l'obiettivo, la priorità, la tracciabilità sui test coverage item, le precondizioni (per l'esecuzione), gli input e i risultati attesi

48

48

Test Specification: Test Procedure Specification

- ❑ Descrive i casi di test in ordine di esecuzione, insieme alle azioni che possono essere richieste per il set up delle necessarie precondizioni e ogni post-execution wrap-up activity
- ❑ Una test procedure specification include un identificatore unico, obiettivo, priorità, azioni di start-up, lista ordinata di test case, relazioni (dipendenze) con altre procedure, azioni necessarie per interrompere l'esecuzione e quelle successive al completamento della procedura

49

49

Test Data and Test Environment Requirements and Readiness Reports

- ❑ Test Data Requirements: descrive i test data necessari per l'esecuzione delle test procedure definite nella test procedure specification
- ❑ Test Environment Requirements: descrive le proprietà del test environment necessarie per eseguire le procedure di test definite nella test procedure specification
- ❑ Test Data Readiness Report: descrive il compimento di ogni test data requirement
- ❑ Test Environment Readiness Report: descrive il compimento di ogni test environment requirement

50

50

Test Execution Documentation

- ❑ Actual results: sono dei record dei risultati dell'esecuzione di un test case. Sono confrontati ai risultati attesi per determinare il risultato del testing
- ❑ Test results: è un record di se o meno uno specifico test case è passato (gli actual result corrispondono ai risultati attesi) o è fallito, o se non è stato possibile eseguire il test case.
- ❑ Test Execution Log: memorizza i dettagli dell'esecuzione di uno o più test procedure come una serie di eventi, che possono essere descritti in una lista, in una tabella, in un documento, in un DB, o in un tool dedicato.
 - ✓ Per ogni evento include un identificatore unico, data e ora, una descrizione e l'impatto se rilevante

51

51

Test Incident Report

- ❑ E' un documento che descrive ogni evento occorso in un processo di testing che richiede altre e più approfondite analisi ed investigazioni;
 - ✓ *This may be called a problem, test incident, defect, trouble, issue, anomaly, or error report*
- ❑ Per ogni incident sono descritti data e ora (come nel test log), l'originator (la persona che ha riportato l'incident), il contesto, la descrizione dettagliata (ai fini della replicabilità), un assessment della gravità e della priorità, rischi e stato dell'incident.

52

52

Documenti di testing e Qualità

- ❑ I documenti di testing stanno assumendo un ruolo di straordinaria importanza nell'era della qualità
 - ✓ i documenti di testing sono una essenziale sorgente di informazioni per la valutazione e la certificazione di qualità;
- ❑ La produzione e la manutenzione di tali documentazione è stata ritenuta tediosa e costosa nel passato (... un alibi per giustificare la totale assenza di ben definiti processi di testing?)

53

Organizzazione dei documenti di testing

- ❑ Le tecnologie hardware e software odierne fanno cadere tali alibi
 - ✓ l'esistenza di supporti automatici per la memorizzazione e gestione automatica di tale documentazione la rende obbligatoria in ogni ambiente di produzione, manutenzione ed evoluzione;
 - ✓ la organizzazione e la implementazione di una **REPOSITORY DI TESTING** è ormai un supporto indispensabile;
- ❑ I documenti e la loro struttura vanno adattati ('tailored') all'ambiente e alla maturità dei processi di testing che in esso si sviluppano.

54