



# REGIONE BASILICATA

## UFFICIO AMMINISTRAZIONE DIGITALE

### Piano dei Test "Id Sistema"

<b>Fornitore:</b>	[digita qui il nome del fornitore]
<b>Prodotto:</b>	[digita qui il nome del prodotto software]
<b>Tipo Intervento:</b>	[scegli il tipo di intervento]
<b>Titolo Intervento:</b>	[digita qui il titolo dell'intervento]
<b>C.I.G./C.U.P.:</b>	[digita qui il codice unico di progetto]

### CONTROLLO DEL DOCUMENTO

APPROVAZIONI			
	Data	Autore	
Redatto da:			
Approvato da:		Ing. Vincenzo Fiore	
VARIAZIONI			
Versione prec.	Data	Autore	Paragrafi modificati
0.0			
0.1			
0.2			
0.3			
DISTRIBUZIONE			
	Copia n°	Destinatario	Locazione



## Indice

1. Introduzione .....	3
1.1 Scopo del Documento .....	3
1.2 Definizioni ed Acronimi .....	3
1.3 Riferimenti .....	3
1.4 Overview.....	3
3. Processo dei Test .....	4
3.1 Generalità.....	4
4. Test Funzionali .....	6
4.1 Ambiente Operativo .....	6
4.1.1 Ambiente Hardware.....	6
4.1.2 Ambiente Software .....	6
4.1.3 Descrizione Strumenti Operativi di Supporto .....	7
4.2 Base Dati di Test.....	7
4.3 Progettazione.....	8
4.3.1 Matrice Requisiti di Business .....	8
4.3.2 Requisiti e Script di Test.....	8
4.4 Esecuzione – Esito dei Test.....	12
4.4.1 Rapporto di Esecuzione .....	12
7. Test non Funzionali .....	13
7.1 Test Prestazionali .....	13
7.2 Test di Sicurezza .....	14
7.3 Test di Stress.....	14
8. Test di non Regressione .....	16
9. Test di Sistema – Integrazione.....	17



---

## 1. Introduzione

[La fase di testing, vista solo come fardello oneroso delle attività di controllo, deve essere invece considerata parte integrante della fornitura, e soprattutto fase primaria del processo di miglioramento continuo della qualità. Il test è la parte più costosa fra le attività di controllo della qualità, ma è anche la parte visibile delle verifiche e delle validazioni realizzate dalla DA per assicurare la qualità ed è l'ultima barriera prima del concretizzarsi del rischio.

La fase di test è distinta e precedente la fase di collaudo di un S.I.; ha l'obiettivo di rilevare ed intervenire su eventuali problematiche prima del collaudo del prodotto presso il cliente.]

---

### 1.1 Scopo del Documento

[Lo scopo di questo documento è quello di evitare che il test diventi collaudo, sottraendoci in questo modo la possibilità di avere una misura della qualità del software in tempi non utili. ]

---

### 1.2 Definizioni ed Acronimi

[Lista e descrizione delle definizioni e degli acronimi.]

---

### 1.3 Riferimenti

[Riferimenti bibliografici, documenti, articoli, siti web di riferimento.]

---

### 1.4 Overview

[Questa sezione riporta cosa il documento contiene e come sono organizzati i contenuti.]



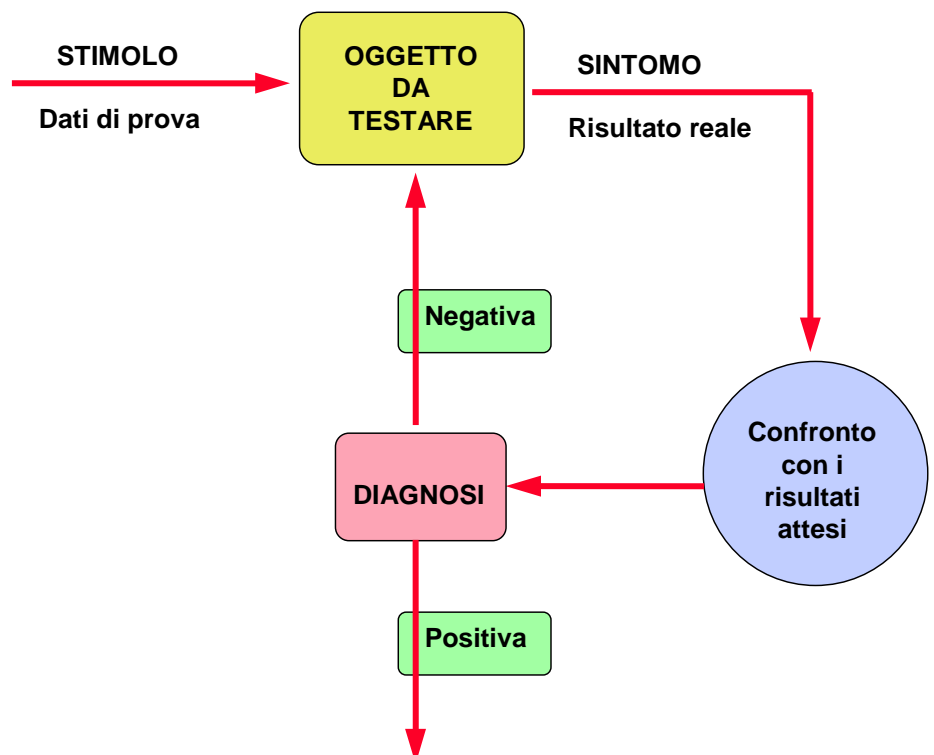
### 3. Processo dei Test

#### 3.1 Generalità

Il processo di test prevede una serie di attività che sono dirette a sollecitare le varie aree dell'applicazione in modo da verificare che i comportamenti di quest'ultima siano conformi alle aspettative.

Le attività previste per disporre il processo di test sono:

- progettazione, in questa fase si procede all' identificazione delle funzionalità, dei requisiti e delle condizioni che si vogliono esercitare per giudicare l'oggetto di test adeguato alle aspettative
- definizione e popolamento degli ambienti, fase in cui si definiscono l'ambiente operativo e si predispongono la base dati in cui si eseguiranno i test
- esecuzione, fase in cui si effettuano le prove, atte a verificare la conformità dei risultati a quanto definito nei documenti di progettazione
- evidenza dei risultati, fase in cui avviene la registrazione dell'esito delle esecuzioni e dello stato delle anomalie
- trattamento delle anomalie, fase in cui le anomalie vengono classificate secondo i livelli di gravità





L'input da fornire all'oggetto da sottoporre a test (lo stimolo) è costituito dai dati di prova progettati; l'output (il sintomo), è costituito dai risultati (dati e/o comportamenti) ottenuti dalla prova di esecuzione effettuata.

Il confronto fra i risultati ottenuti ed i risultati attesi a fronte dei dati di prova progettati conduce alla diagnosi che può consistere nell'accettazione dell'oggetto sottoposto a test oppure nella segnalazione di eventuali malfunzionamenti (non conformità rispetto ai requisiti).



## 4. Test Funzionali

[In questa sezione si tratterà in maniera puntuale il processo di testing atto a valutare i requisiti funzionali di un S.I. I test non funzionali, di sistema, e di non regressione devono seguire lo stesso iter procedurale.]

### 4.1 Ambiente Operativo

[In questo paragrafo sarà descritta la sequenza di attività da eseguire per predisporre l'ambiente di collaudo in cui eseguire il test.]

#### 4.1.1 Ambiente Hardware

[Si riportano di seguito le configurazioni hardware necessarie per lo svolgimento delle attività di esecuzione dei test.]

Ambiente Hardware			
Componente (Processore)	Quantità	Marca e Modello	Ram Mb

#### 4.1.2 Ambiente Software

[Si riportano di seguito le configurazioni software necessarie per lo svolgimento delle attività di esecuzione dei test.]

Ambiente Software		
Componente	Quantità	Versione/Release



#### 4.1.3 Descrizione Strumenti Operativi di Supporto

Strumenti Operativi di Supporto		
Strumento	Descrizione	Versione/Release

#### 4.2 Base Dati di Test

[Caratteristiche essenziali di un buon test sono completezza e possibilità di ripeterne l'esecuzione; a tal fine occorre specificare i criteri attuati (eseguiti) per la predisposizione della base dati, riportando informazioni quali: fonti della base dati, modalità di generazione, ed infine la stima o il valore effettivo delle dimensioni della base dati di test.]

Base Dati di Test			
Fonti della Base Dati	Modalità Generazione Base Dati	Stima Dimensioni Base Dati di Test	Valore Effettivo Dimensioni Base Dati di Test



## 4.3 Progettazione

La fase di “progettazione” consiste nell’identificazione di funzionalità, requisiti e condizioni da esercitare per valutare l’elemento oggetto di test.

### 4.3.1 Matrice Requisiti di Business

[Dopo aver individuato le funzionalità da sottoporre a test, occorrerà valutare se la funzionalità è di tipo elementare , o in caso contrario, occorrerà decomporre in funzionalità elementari.]

Matrice Requisiti di Business (Scomposizione Funzionale del Progetto)		
Macro Funzionalità	Funzionalità Principali	Funzionalità Elementare

### 4.3.2 Requisiti e Script di Test

[Per ciascuna funzionalità elementare occorrerà identificare un requisito di test e sviluppare uno script di test. Uno script test consiste in una sequenza dettagliata di azioni da svolgere, nell’identificazione dei dati da fornire in input, e nell’individuazione dei risultati attesi.

Sarà inoltre opportuno prendere in considerazione sia casi positivi, che servono a testare che l’applicazione faccia ciò che deve fare; sia casi negativi, atti a verificare che l’applicazione non faccia nulla che non deve fare. ]





Test Planning											
Requirement Test Funzionale								Test Case			
Id Requisito	Id	Macro Funzionalità	Id	Funzionalità Principale	Id	Funzionalità Elementare	Livello Gravità	Id Test Case	Titolo Test Case	Livello Gravità	Id Test Script

Poiché l'amministrazione intende disporre di un patrimonio di test per il futuro, ogni script di test deve essere:

- entrocontenuto (non deve dipendere da altri test per essere utilizzato)
- dettagliato
- prevedere pre e post condizioni
- prevedere eventuali ricicli



**N.B.** Il livello di gravità è da intendersi associato al verificarsi di un'anomalia della funzionalità che si sta testando.

Le anomalie vengono classificate secondo quattro livelli di gravità. La tabella che segue ne illustra i criteri di valutazione.

Livelli Gravità Anomalie	
Livello Gravità	Anomalie
Alta	L'anomalia risulta bloccante per funzioni dell'applicazione, oppure provoca contaminazioni sulle informazioni archiviate, oppure provoca rallentamenti evidenti per servizi vitali o critici
Media	L'anomalia provoca rallentamenti evidenti per servizi non vitali né critici, oppure presenta gravi carenze di usabilità, oppure provoca il blocco della funzione se vengono immessi dati o compiute operazioni scorrette
Bassa	L'anomalia provoca lievi problemi di usabilità, oppure provoca comportamenti non previsti dell'applicazione se vengono immessi dati o compiute operazioni scorrette ma non ne provoca il blocco né gravi difficoltà funzionali
Warning	L'anomalia non provoca problemi, ma vengono evidenziati ambiti di miglioramento



Casi di Test										
Id Test Script	Nome Test Script	Tipo Test	Operazioni Preliminari	Step	Descrizione (Action)	Dati Input	Modalità Esecutive	Risultato Atteso	Modalità di Verifica	Obiettivo del Test



## 4.4 Esecuzione - Esito dei Test

L'esecuzione del test è l'attività attraverso la quale si esercitano le prove, per verificare la conformità dei risultati a quanto definito nei documenti di progettazione.

### 4.4.1 Rapporto di Esecuzione

L'esito dei test viene registrato in un prospetto, il rapporto di esecuzione che evidenzia l'esito delle esecuzioni e lo stato delle anomalie

Rapporto di Esecuzione						
Id Test Script	Nome Test Script	Esecutore	Data Inizio	Data Fine	Difetto/Anomalia	Note

Eventuali anomalie vengono descritte in un apposito prospetto

Registro Anomalie						
Id Test Script	Nome Test Script	Identificativo Difetto	Livello Gravità	Descrizione Difetto	Data Comunicazione	Stato

L'applicazione, o la parte di essa, sulla quale "pende" un rapporto di non conformità viene prelevata dall'ambiente di test, riportata nell'ambiente di sviluppo e rilavorata; un successivo ciclo di test potrà verificare l'avvenuta rimozione delle anomalie ed emettere un rapporto di conformità.



## 7. Test non Funzionali

[L'insieme dei test non funzionali contiene categorie di test finalizzati a verificare funzionalità non applicative e relative ad aspetti e/o requisiti di sicurezza, di prestazione, di usabilità.

Categorie esemplificative sono:

- Test prestazionali, atti a verificare che il sistema soddisfi i requisiti prestazionali richiesti
- Test di sicurezza, atti a verificare che il sistema realizzato garantisca i requisiti di sicurezza richiesti
- Test di compatibilità, atti ad assicurare il corretto funzionamento del sistema realizzato su differenti piattaforme hardware, di sistema operativo, di browser, ecc. oppure la perfetta compatibilità del sistema con gli altri prodotti con i quali è correlato
- Test di usabilità, atti a verificare che il sistema realizzato soddisfi tutti i requisiti di usabilità richiesti, come la facilità di comprensione dell'interfaccia o l'adeguatezza dei contenuti delle schermate. Questa tipologia di test permette di identificare criticità e colli di bottiglia dell'interfaccia (da correggere in fase di design), e di capire come l'utente si muove e ragiona per scoprire eventuali difficoltà di cui tener conto in fase di progettazione
- Test di stress o di carico del sistema, necessari se si è in presenza di progetti critici, atti a garantire la capacità del nuovo sistema di funzionare correttamente anche quando si raggiungono i suoi limiti in termini di quantità di dati trattati o di utenti connessi.

La conduzione (metodologia) di tale tipo di test, dall'analisi, alla progettazione, all'esecuzione, è analoga a quella già descritta per i test funzionali, potrebbe essere utile attribuire a tale tipo di test un identificativo progressivo del tipo TNF.xxxx, aggiungendo una descrizione del test, una breve descrizione delle modalità e/o degli strumenti con cui si ritiene di poter eseguire il test.]

### 7.1 Test Prestazionali

[Lo scopo del performance testing non è di rilevare errori nell'applicazione, ma di eliminare potenziali colli di bottiglia e stabilire una baseline prestazionale per i futuri test di regressione. Condurre test di questo tipo è equivalente all'esecuzione di un processo controllato di misurazione e analisi. Un insieme chiaramente definito di "cosa ci si aspetta" è essenziale per un test significativo. Ad esempio, per un'applicazione web sarebbe necessario conoscere almeno:

- carico atteso in termini di utenti concorrenti o connessioni http
- tempi di risposta considerati accettabili

Occorre costruire e pesare requisiti prestazionali per ogni funzione elementare, in particolare le misure necessarie da effettuare sono:

- carico massimo previsto
- tempo di risposta massimo accettabile

Affinché un collaudo prestazionale abbia buona probabilità di successo l'applicazione deve essere sufficientemente stabile e l'ambiente di test stabile e reale.

Il test dovrà essere effettuato impiegando strumenti specifici che:

- consentano la creazione di casi di test capaci di stimolare il nuovo sistema simulando massicci accessi tali da riprodurre scenari operativi reali



- consentano la parametrizzazione dei casi di test
- rendano disponibili informazioni relative alle esecuzioni dei test, e permettano la generazione di reports personalizzabili contenenti indicazioni su scenario, test eseguito misurazioni effettuate (performances)

Tali reports dovranno essere allegati al piano dei test.

---

## 7.2 Test di Sicurezza

Non esistono metodi standard per collaudare la sicurezza di un'applicazione. Inoltre, sono disponibili pochi strumenti per collaudare in modo approfondito tutti gli aspetti della sicurezza. La tecnica principale dei test della sicurezza consiste nel tentare di violare i controlli di protezione predefiniti. Questa tecnica assicura che i meccanismi di protezione impediscano l'accesso indesiderato al sistema.

Esistono due tipi di attacchi: interni ed esterni. Per attacco interno si intende un attacco messo in atto da un dipendente ed alcune delle minacce più pericolose per l'infrastruttura di un'organizzazione sono costituite dagli attacchi provenienti dalla rete interna. Gli attacchi interni che potrebbero causare potenzialmente i danni maggiori sono quelli risultanti dalle attività di persone a cui sono assegnati i privilegi più alti, ad esempio gli amministratori di rete.

Un attacco esterno, invece, proviene dall'esterno dell'organizzazione.

Possibili categorie di attacco sono:

- tentativi di compromissione delle credenziali
- sfruttamento dei punti di vulnerabilità
- accesso a computer non autorizzato

Possibili esempi di test potrebbero essere:

- attacco con dizionario
- inviare un elevato numero di richieste continue al sistema, negando il servizio ad altri
- causare volontariamente errori di sistema per tentare di accedervi durante il ripristino
- esplorare dati non protetti per individuarne la chiave di accesso

---

## 7.3 Test di Stress

Tale tipo di test non è necessariamente associato ad un requisito funzionale ed è normalmente eseguito con l'utilizzo di prodotti specifici e da specialisti del test, pianificando le attività con gli sviluppatori. E' opportuno indicare la necessità di tale tipo di test e dettagliarne le modalità di esecuzione. I test di stress aumentano costantemente il carico del sistema oltre il carico massimo progettato, fino a quando non si verifica un errore nel sistema. Questo tipo di test ha una duplice funzione:

- determinare il comportamento del sistema in caso di errore provocato da un sovraccarico del sistema che ha comportato una perdita di dati o del servizio per gli utenti
- sottoporre il sistema a un carico eccessivo in modo da rilevare eventuali difetti che potrebbero rimanere celati in condizioni normali.



I test di stress sono particolarmente indicati per sistemi e-commerce e applicazioni web, infatti sono questi i sistemi che mostrano spesso un calo significativo delle prestazioni in condizione di rete sovraccarica. Quindi lo stress test ha lo scopo di provare a “rompere” il sistema sotto test sovraccaricando o sottraendogli risorse. Lo scopo principale di queste attività è verificare se il sistema andato in “fault”, successivamente è in grado di recuperare in maniera “indolore”, cioè è un sistema fault-tolerant.

A differenza del “performance test” che richiede un ambiente controllato e misure ripetibili, lo stress test provoca caos e imprevedibilità.

Riferendoci ad un’applicazione web, alcuni esempi di come stressare un sistema sono:

- raddoppiare il numero di utenti/connessioni HTTP previste in baseline
- in maniera casuale spegnere e riavviare porte dei switch/router di rete che collegano i server
- mettere offline il database, e farlo ripartire
- eseguire processi che consumano risorse (CPU, memoria, disco, rete) sui web-server

La lista può essere ovviamente arricchita. Comunque, lo stress test non è fatto al puro scopo di far andare in crash il sistema, ma piuttosto deve servire per osservare come il sistema reagisce alle failure. E osservare se al riavvio, è in grado di recuperare dall’ultimo stato corretto; se visualizza messaggi di errore comprensibili all’utente; se la sicurezza del sistema è compromessa a causa di un failure inatteso.



---

## 8. Test di non Regressione

I test di non regressione servono a verificare la correttezza delle funzionalità impattate dagli interventi effettuati.

Tale tipo di test è importante perché è il processo di accertamento che assicura che i programmi continuino a funzionare senza problemi dopo aver apportato i nuovi cambiamenti.

Anche per tale tipo di test la metodologia da adottare è analoga a quella usata per i test funzionali.

Potrebbe essere utile ed opportuno attribuire ad ognuno di essi un identificativo di test progressivo TNR.xxxx, descrivere il test e la modalità con cui si vuole eseguire, facendo riferimento alle funzionalità oggetto di test.





---

## 9. Test di Sistema - Integrazione

E' il processo di test applicato al sistema finito e completo che deve essere convalidato rispetto ai suoi requisiti funzionali.

Infatti è d'uopo al termine delle singole sessioni di test, eseguire un "Test di Sistema", come verifica conclusiva del corretto funzionamento del sistema in rilascio, considerando integrate tutte le sue diverse componenti. Può accadere che anche se i singoli moduli sono corretti, il sistema ottenuto integrandoli potrebbe non esserlo.

Nel test di sistema è incluso, oltre all'esecuzione del piano completo dei test funzionali, anche l'effettuazione dello stress testing e del performance test.

Per collaudare un sistema completo si deve utilizzare il software in modo il più possibile simile a come verrà utilizzato nella normale operatività