



D4 - Rapporto Tecnico contenente il test e la sperimentazione delle nuove tecnologie proposte

**DRAGO – Digitalizzazione reperti
archeologici garganici by open-source**

Prima emissione: 01/06/2015

Distribuzione: PUBBLICA

INDICE GENERALE

1	SCOPO	3
2	VALIDITÀ.....	3
3	SISTEMA DI GESTIONE DEL COLLAUDO.....	4
	3.1 Versioning.....	4
	3.2 Gestione Anomalie	5
	3.3 Sviluppo Prototipale.....	5
4	GESTIONE DEI TEST INTERNI.....	6
	4.1 Attività di test di Unità	7
	4.2 Attività di test di integrazione	7
	4.3 Attività di test di performance e stress	8
5	GESTIONE DELLA SPERIMENTAZIONE	8
	5.1 Premessa	8
	5.2 Attività di sperimentazione	9

1 SCOPO

Questo documento ha lo scopo di descrivere il test e la sperimentazione delle nuove tecnologie proposte.

2 VALIDITÀ

Le indicazioni contenute nel presente documento hanno validità unicamente per il Progetto DRAGO – Digitalizzazione reperti archeologici garganici by open-source.

3 SISTEMA DI GESTIONE DEL COLLAUDO

3.1 Versioning

La gestione del collaudo, richiede la messa a punto di una struttura organizzativa efficiente per permettere di monitorare ogni singolo componente software di ogni installazione e raccogliere ogni istanza di anomalia, tale da tracciare, nelle singole installazioni, la corretta versione delle componenti e quindi la conseguente risoluzione degli errori riscontrati nel codice.

Man mano che vengono prodotti nuovi componenti dai partner del progetto DRAGO, con il piano di attività illustrato nel piano di attuazione, questi vengono sottoposti a collaudo da parte dei tecnici. In queste fasi, benché non venga arrestata l'attività di sviluppo dei componenti successivi, il personale di tutti i partner provvede all'assistenza dei componenti in collaudo al fine di garantire la prosecuzione dello stesso.

Data la natura modulare e altamente complessa del sistema, con l'introduzione di nuove componenti non si è potuto prescindere, in una fase di collaudo, dalla realizzazione di un insieme di procedure e dalla predisposizione di opportuni strumenti software che permettano il versioning del prodotto.

È stato necessario pertanto tracciare ogni versione del software onde permettere al programmatore di ripristinare, su ciascuna macchina di sviluppo, la versione del software e la base dati dell'utente che individua un'anomalia, al fine di garantirgli il massimo supporto.

La gestione delle versioni così codificate è stata implementata a forma di albero, in modo da applicare lo stesso criterio per ciascun componente. Vista la presenza di diversi programmatori che lavorano sullo stesso prodotto, è stato necessario predisporre un repository unico dei sorgenti nelle varie versioni di tutti i componenti del software e dei moduli del server, per impedire disallineamento dei dati e sovrascrittura di codice.

Una serie di funzionalità sono state messe a disposizione da uno strumento software che gestisce un repository con alberatura del codice, suddiviso per versioni, in modo che la creazione di una nuova versione che risolve determinati errori o che supporta determinate migliorie, sia data dal merging del codice dei singoli componenti dei diversi programmatori che sono interessati e dalla base dati opportuna con i vari aggiornamenti delle informazioni prodotte.

Al programmatore è stato dato a disposizione uno strumento in grado di scaricare dal repository una qualunque versione del prodotto, in modo da garantire un'assistenza adeguata e mirata alla versione dell'utente e fornire quindi un servizio più efficiente.

La gestione del versioning e dell'attività di testing richiede anche la necessità di predisporre un ambiente di testing in termini di hardware. Un PC predisposto al test deve essere configurato in modo tale da permettere di riprodurre una determinata condizione in maniera esatta, ovvero di riprodurre una determinata anomalia.

Un sistema che reinstalli il software e ripredisponga l'ambiente in maniera "pulita" diventa quindi una condizione necessaria per la realizzazione del test.

3.2 Gestione Anomalie

Per la tracciabilità delle anomalie e per il collegamento al sistema di versioning appena illustrato, è stato messo a disposizione un data base per la gestione delle chiamate degli utenti, per la gestione degli interventi e per il rilascio delle versioni successive dei componenti.

Quando l'utente segnala un'anomalia per telefono, per e-mail o per via diretta, con lo strumento appena descritto, il programmatore apre una chiamata, alla quale si attribuisce un numero di chiamata e un numero di anomalia e una priorità (bassa, media e alta). La chiamata farà partire un intervento che potrà essere risolto direttamente presso l'utente o con il rilascio di un componente con nuova versione. Se l'intervento di risolve presso l'utente, la chiamata viene chiusa. Se occorre invece la realizzazione di attività di sviluppo per la soluzione delle anomalie, viene effettuata la modifica dietro preventivo parere favorevole del coordinatore e risolta in base alla priorità, oppure viene ricompilata una nuova versione del componente e questo viene rilasciato immediatamente o ancora, nel caso delle anomalie con più bassa priorità, si raggruppano le stesse e si rilascia il componente successivamente.

Quando opportuno e sempre a seguito dell'inserimento di nuove funzionalità, si è realizzata una nuova versione della piattaforma in un pacchetto di installazione al quale verrà data una versione generale con un numero di release.

3.3 Sviluppo Prototipale

Lo sviluppo software che si è adottato e quello definito per prototipi che è specifico per applicazioni WEB come portali o siti specifici ove il ciclo di sviluppo è detto iterativo, o "evolutionary", piuttosto che un tradizionale ciclo di sviluppo a cascata.

Il metodo *evolutionary* presenta la caratteristica di essere orientato al soddisfacimento del Progetto, in particolare:

- i cicli vengono pianificati in base a funzionalità visibili al committente e all'utente finale e non in base a considerazioni legate alla suddivisione del sistema;
- l'utente può ricevere una nuova versione del prodotto ad ogni ciclo e può verificare tempestivamente che il prodotto si stia evolvendo in base alle sue aspettative, fornendo feedback che possono influenzare tutti i cicli seguenti.

D4 - Rapporto Tecnico contenente il test e la sperimentazione delle nuove tecnologie proposte

Adottando il ciclo di sviluppo *evolutionary*, il progetto è stato scomposto in moduli di dimensioni limitate, ciascuno costituita da una o più funzionalità tra quelle che il sistema deve svolgere. Pertanto, al completamento di ogni modulo, è stato rilasciato un sottoinsieme utilizzabile del sistema.

All'interno di ciascun ciclo, l'architettura funzionale preliminare è stata perfezionata e sono state eseguite le attività di progettazione, implementazione e test; quindi il sistema è stato rilasciato all'utente. Quest'ultimo ha fornito un feedback sul sistema che è stato utilizzato dai responsabili del progetto al fine di applicare le eventuali modifiche. Questo approccio viene applicato fino alla consegna definitiva del sistema.

Al fine di garantire un livello qualitativo eccellente alle proprie realizzazioni si sono poste particolari attenzioni alle attività di test. Obiettivi primari di tali attività sono i seguenti:

- validare l'usabilità del prodotto: oltre alla facilità d'uso, la navigazione tra le pagine del sito e la consistenza delle informazioni in esso contenute
- validare le funzionalità del sistema:
- Funzioni di usabilità: verificare che esse siano supportate dal sito, in particolare validando il maggior numero di scenari di utilizzo del sistema
- Navigazione: controllare il corretto comportamento di tutte le funzioni di navigazione
- Operazioni di base: verificare i controlli e i comandi anche rispetto alla navigazione del sistema
- Verificare che le condizioni di errore e "failure" del sistema siano gestiti correttamente
- Verificare le performance del sistema:
- Concorrenza di utilizzo
- Tempi di risposta
- Performance all'aumentare del carico
- Capacità: picchi di carico
- Sistema sotto stress
- Verificare la compatibilità del prodotto con i diversi Sistemi Operativi e browser supportati (bisogna decidere per quali browser validare la compatibilità: sicuramente verrà provato su (IE 4.X, 5.0 e 6.0)

4 GESTIONE DEI TEST INTERNI

Le attività di verifica e validazione avvengono in diversi momenti del ciclo di vita del software. I test di unità sono necessari e obbligatori, su richiesta del Project Manager sono stati effettuati i test di integrazione e di performance e stress a seconda delle esigenze del sistema. L'esito dei test ha contribuito a determinare lo spazio di applicazione indicato nel programma di attuazione.

I test non sono stati eseguiti da chi ha sviluppato il singolo modulo software, perché vi deve essere indipendenza tra sviluppatore e tester.

4.1 Attività di test di Unità

In questa fase dopo lo sviluppo ciascun programmatore si è occupato di definire dei test di unità del modulo da lui sviluppato. In particolare ha definito gli obiettivi e i casi di prova per la verifica. Il test è stato definito pensando alle funzionalità sviluppate e come queste sono illustrate nel documento dei requisiti del software, oltre ai documenti di dettaglio ed eventualmente architetturale che gli è stato fornito.

Per ciascuna funzionalità che si vuole sottoporre a test si è richiesta la compilazione di una checklist con tutti gli elementi da controllare. Poi si sono prodotte le check list ovvero una lista di controlli da effettuare per ciascun test.

Le check list per un certo test possono essere più di una e sono state ben dettagliate. Il programmatore ha lanciato queste check list in unità di lancio andando a compilare le singole voci del check e indicando l'esito con il dato che ha provato.

Le unità di lancio sono state anche più di una per ogni check list ad esempio le verifiche sui limiti dello spazio dei dati (oltre un valore massimo o al di sotto di un valore minimo), inserendo il dato corretto e inserendo il dato errato.

Ogni checklist aveva un **critério di accettabilità** da inserire nelle note in modo da poter stabilire chiaramente quando l'esito è positivo o negativo.

Il project manager ha verificato i singoli test, se questi sono tutti passati e ha proceduto ad assegnare al responsabile dei test la fase del test di integrazione quando era necessario.

Se i test non erano esaustivi perché non erano state testate determinate funzionalità o non erano stati completati in modo esaustivo le checklist, il project manager ha richiesto al programmatore di integrare la fase di validazione con ulteriori test prima di passare alla fase di test di integrazione.

Qualora delle unità di lancio non siano passate correttamente, il tester ha aperto sul sistema di gestione delle chiamate una anomalia come indicato nel paragrafo precedente descrivendo con chiarezza il problema e il riferimento all'esito del test, il project manager ha trattato la gestione della anomalia assegnando l'attività al programmatore. Il programmatore ha gestito quindi l'anomalia, riportando la descrizione della soluzione e chiedendo al tester di effettuare nuovamente una unità di lancio. In un ciclo che prosegue fino alla risoluzione di tutte le anomalie.

4.2 Attività di test di integrazione

Il test di integrazione avviene sul server di test, ciascun programmatore ha installato il proprio modulo configurandolo sul server di test ed è stato reso disponibile al responsabile dei test.

Successivamente il responsabile del test ha proceduto con la compilazione dell'elenco dei test, dei singoli test, delle checklist e delle unità di lancio come indicato nel paragrafo precedente, però specificatamente solo per la verifica delle integrazioni tra le parti.

Il project manager ha verificato l'esito dei test e quando l'ha ritenuto necessario ha chiesto di integrarli con ulteriori test.

Qualora un test non è passato si è proceduto a gestirli come indicato nel paragrafo precedente. Al termine delle modifiche si è andati avanti con il lancio dei test da parte del responsabile dei test.

4.3 Attività di test di performance e stress

Il test di performance e stress avviene sul server di test a seguito del test di integrazione e se quest'ultimo ha avuto successo.

Il responsabile dei test con software opportuno ha proceduto alla compilazione dell'elenco dei test, dei singoli test, delle checklist e delle unità di lancio come indicato nel paragrafo precedente e ha verificato il comportamento del sistema nei casi estremi legati al numero di utenti collegati contemporaneamente alle sessioni attive in un certo istante e alla dimensione del database.

Il project manager ha verificato l'esito dei test e quando l'ha ritenuto necessario ha richiesto di integrarli con ulteriori test e se tali test non sono passati sono stati gestiti come indicato nel paragrafo *delle attività di Test di unità*. Al termine delle modifiche si è andati avanti con il lancio dei test da parte del responsabile dei test.

5 GESTIONE DELLA SPERIMENTAZIONE

5.1 Premessa

In qualità di Global System Integrator, il coordinatore del progetto si è proposto di rispondere ai requisiti di qualità del progetto e rispetto dei tempi, conciliando i requisiti di *affidabilità* di tutta la soluzione implementata con gli obiettivi strategici di riferimento e i *vincoli* implementativi (tempi e risorse).

Al fine di rendere raggiungibile questo obiettivo, si è adottato il metodo di gestione progetti fondato sul Capability Maturity Model (CMM) del Software Engineering Institute (SEI), Carnegie Mellon University, ossia il più affermato modello di maturità dei processi ICT a livello internazionale per progetti complessi che fanno uso di tecnologie avanzate.

Tale modello consente di massimizzare il raggiungimento degli obiettivi di progetto in termini di:

D4 - Rapporto Tecnico contenente il test e la sperimentazione delle nuove tecnologie proposte

- rispetto dei tempi pianificati;
- contenimento dei costi in accordo con quanto stimato;
- rispetto degli standard di qualità prefissati.

Il raggiungimento di questi obiettivi dipende da un'accurata gestione che consenta:

- la conoscenza dei requisiti;
- la consapevolezza e il controllo dei rischi;
- la trasparenza dei risultati;
- il controllo sul processo e il feedback agli utenti finali;

Tale gestione accompagna tutto il ciclo di sviluppo della soluzione, modulando gli impegni in funzione delle effettive necessità delle varie fasi di lavoro e delle esigenze di progetto.

L'applicazione di un metodo rigoroso di project management risulta ancora più determinante nel progetto specifico, il quale si caratterizza per una sensibile complessità data dal coinvolgimento di diversi attori, dalla necessità di coordinamento di diversi pacchetti di lavoro fortemente interrelati, dai brevi orizzonti temporali all'interno dei quali devono essere raggiunti gli obiettivi.

5.2 Attività di sperimentazione

La sperimentazione per il progetto DRAGO, è avvenuta attraverso il rilascio di prototipi fruibili via WEB e/o tablet agli utenti dei gruppi di lavoro.

Si è raggiunto un numero di circa 100 utenti. Per alcuni di loro è stato importante procedere ad una fase di training, mentre per altri non è stato necessario perché hanno utilizzato il sistema ed hanno appreso in modalità da autodidattica.

Tale approccio ha favorito una valutazione anche dell'immediatezza e friendliness dell'interfaccia grafica utilizzata.

Ciascun utente ha potuto navigare sul museo virtuale, sulle scansioni 3D, sulle immagini dei reperti, sull'enciclopedia, hanno potuto usare l'e-book, vedere i video, ascoltare le interviste e direzionarsi in un ambiente immersivo e virtuale con una certa facilità.

Alcuni hanno coinvolto i propri parenti e/o conoscenti emigrati in altri paesi fuori dall'Italia, che hanno con piacere visitato le bellezze artistiche dei siti dell'area garganica e alto tavoliere.

D4 - Rapporto Tecnico contenente il test e la sperimentazione delle nuove tecnologie proposte

Potendo fruire del sistema tramite WEB è stato possibile, non solo renderlo disponibile a livello internazionale ma anche di poter lasciare aperta la possibilità di utilizzo senza limiti di tempo e/o altre restrizioni.

Le eventuali modifiche apportate, sono state realizzate sulla piattaforma che automaticamente, senza necessità di ricompilazione del codice, si è resa disponibile su Internet. Questo è stato possibile, senza riavviare il server o l'applicazione, perché sia i dati che la logica implementativa risiede tutta in modo open sul database.

La somministrazione di un questionario a ciascuno di loro ha permesso di acquisire delle informazioni utili al riscontro sulla gradevolezza, sulla semplicità di utilizzo e sulla completezza e interesse del sistema.