Configuration Management

Versione: 0.6 ***Data:*** 21/02/2019
Autore: Pellizzari Luca ***Responsabile:*** Di Benedetto Gianluca

Indice

Nome documento	Versione	Autore	Data
GDPRPrj_CM_v0.1	0.1	Pellizzari Luca	27/11/2018
GDPRPrj_CM_v0.2	0.2	Pellizzari Luca	30/11/2018
GDPRPrj_CM_v0.3	0.3	Pellizzari Luca	13/12/2018
GDPRPrj_CM_v0.4	0.4	Pellizzari Luca	06/01/2019
GDPRPrj_CM_v0.5	0.5	Pellizzari Luca	17/01/2019

1. Organizzazione del team

Il team GL^3 è composto da tre membri: Baradel Luca, Di Bendetto Gianluca, Pellizzari Luca.

Il team si riunisce due volte a settimana:

1) una subito dopo l'incontro con il cliente per fare un'attività di brainstorming quindi per cercare di chiarire gli eventuali dubbi e per individuare gli obiettivi comuni del lavoro;
2) una precedente al prossimo incontro con il cliente per implementare il lavoro pianificato nell'incontro precedente.

Nella prima delle due sedute inoltre si pianifica il lavoro individuale che i membri del team dovranno svolgere entro la data prevista per il secondo incontro in modo da poter discutere (ed eventualmente modificare) i risultati ottenuti prima di mostrarli al cliente.

Il conduttore scelto per il gruppo è Gianluca Di Benedetto.

Gli altri ruoli dei membri del team sono i seguenti:

- Baradel Luca: documentazione, gestione versioni, database;
- Di Benedetto Gianluca: responsabile del CM, dati di test e relativa documentazione;
- Pellizzari Luca: codice, change request, software (tool) di supporto.

Lo strumento scelto dal team per gestire i file relativi al progetto è GitHub; inoltre è stato creato un gruppo Whatsapp per accordarsi sulle date degli incontri e per ulteriori comunicazioni.

2. Identificazione item

Gli item identificati in questa prima fase del progetto sono i seguenti:

1) Documentazione:
    a. Documento dei requisiti;
    b. Analisi dei requisiti;
    c. Casi d'uso UML;
    d. Manuale utente;
    e. Verbale di interazione con il cliente;
    f. Documento di progetto;
    g. Offerta in relazione al mandato;
    h. Configuration Management;
    i. Documento di avanzamento del progetto;
    j. Ambito di lavoro;
    k. Documento di test sui requisiti;
    l. Documento sulla base di dati;
    m. Test di accettazione;
    n. Tempi di lavoro.
2) Change request;
3) Codice;
4) Gestione versioni;
5) Dati di test e relativa documentazione;
6) Database;
7) Software (tool) di supporto.

Di seguito vengono elencate le caratteristiche principali degli item e il modo in cui questi item verranno gestiti dal team di sviluppo.

2.1. Documentazione

Tutta la sezione legata alla documentazione verrà salvata in una repository condivisa fra i membri del team su GitHub, per ogni documento vengono mantenute tutte le sue versioni (che corrispondono a file con indici di versione di primo o secondo livello diversi) in formato pdf e l'ultima versione di ogni documento viene salvata anche in formato docx in modo che sia facilmente modificabile per passare alla versione successiva.

Per ogni documento descritto in questa sezione è presente un template all'interno della repository del team su GitHub nella cartella Documentazione/Templates. In questo modo è possibile per ogni documento avere una versione pronta con intestazione (che comprende: data, autore, responsabile, numero di versione e logo del gruppo) e indice già scritti.

Per ogni documento è presente una breve descrizione in linguaggio naturale e successivamente un elenco delle caratteristiche e delle informazioni minime che il documento deve contenere.

Tutti i documenti identificati da una versione, e non da una data, devono contenere (dopo l'indice se presente) una tabella che contiene la version history (nome documento, autore, data e numero di versione della versione precedente dello stesso documento).

2.1.1. Documento dei requisiti

Descrizione

Il documento dei requisiti è un documento chiave per formalizzare i fabbisogni del cliente relativamente al sistema da sviluppare, in modo non ambiguo. Questo documento può essere usato come contratto tra cliente e sviluppatori. All'interno del documento sono presenti diversi livelli di raffinamento, dal linguaggio naturale ai modelli UML per i casi d'uso che forniscono una rappresentazione intuitiva e sintetica di quali saranno le azioni che un utente potrà compiere utilizzando il sistema che sta per essere sviluppato.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_DocRequisiti_versione.pdf (esempio: GDPRPrj_DocRequisiti_v0.1.pdf);
2) I requisiti vanno classificati in base allo stakeholder (cliente, sviluppatori, utenti finali, eccetera) e divisi in funzionali e non funzionali;
3) Ogni requisito viene scritto in linguaggio naturale strutturato e deve essere inserito nel documento dei requisiti insieme ad un'eventuale motivazione;
4) Un requisito può essere associato ad una priorità che indica il suo valore di business.

2.1.2. Analisi dei requisiti

Descrizione

Gli obiettivi del documento di analisi dei requisiti sono:

1) Definire quali sono le caratteristiche che il sistema dovrà possedere per soddisfare le necessità del cliente;
2) Scrivere un documento completo, preciso, consistente, non ambiguo e comprensibile sia al committente che allo sviluppatore.

Quindi è necessario definire cosa dovrà fare il sistema senza dire come dovrà farlo.

Il punto di partenza per la stesura di questo documento è il documento dei requisiti, quindi prendendo come input i requisiti descritti nel documento, questi vengono analizzati e mappati in specifiche del sistema.

Il documento di analisi dei requisiti costituisce il punto di convergenza di tre diversi punti di vista: utente, cliente e sviluppatore, inoltre fornisce un punto di riferimento per la validazione del prodotto finale.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_AnalisiRequisiti_versione.pdf (esempio: GDPRPrj_AnalisiRequisiti_v0.1.pdf);
2) Il documento di analisi dei requisiti deve contenere, per ogni requisito presente nel documento dei requisiti, una descrizione che ci permette di trasformare il requisito del cliente in un requisito del sistema. In altre parole, ogni azione descritta dal cliente deve essere mappata in un'azione che deve compiere il sistema;
3) Ad ogni documento di analisi dei requisiti deve essere associata una data e un codice che identifica la versione del documento (come descritto nella gestione delle versioni).

2.1.3. Casi d'uso UML

Descrizione

I diagrammi dei casi d'uso sono diagrammi dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Lo scopo è quello di fornire una rappresentazione schematica e intuitiva di quelle che sono le azioni messe a disposizione dal sistema.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà CasiUso_data.pdf (esempio: CasiUso_27-11-2018.pdf);
2) Il file contiene un certo numero di rappresentazioni con sintassi UML che descrivono le azioni messe a disposizione dal sistema;
3) Il file è realizzato con il tool draw.io disponibile online; oltre al file in formato pdf verrà memorizzato anche un file in formato XML in modo che possa essere successivamente esteso/aggiornato.

2.1.4. Manuale utente

Descrizione

Il manuale utente è la pubblicazione tecnica che contiene le informazioni utili ad un corretto utilizzo di un sistema. È diviso in capitoli/sezioni in base agli argomenti e viene mantenuto aggiornato con l'ultima versione del sistema a cui è legato.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_ManualeUtente_versione.pdf (esempio: GDPRPrj_ManualeUtente_v0.1.pdf);
2) Tutte le possibili operazioni che l'utente può svolgere tramite il sistema software che verrà sviluppato devono essere descritte nel manuale;
3) Ogni aggiornamento del sistema che modifica le funzionalità messe a disposizione dallo stesso o che cambia il modo di effettuare alcune operazioni deve essere descritto nel manuale: fra questo e il sistema non devono esserci inconsistenze;
4) Ogni versione (non variante) del sistema deve essere associato ad una versione del manuale.

2.1.5. Verbale interazione con il cliente

Descrizione

Il verbale di interazione con il cliente è un documento che descrive i contenuti discussi durante un incontro con il cliente e le decisioni prese durante l'incontro. Serve per tenere traccia degli accordi presi, delle date dei prossimi incontri e per avere sempre a disposizione i contenuti/temi trattati negli incontri precedenti.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_VerbInteraz_data.pdf (esempio: GDPRPrj_VerbInteraz_2018-11-23.pdf);
2) La struttura deve essere la seguente:
    a. Data (ed eventualmente luogo) dell'incontro;
    b. Presenti;
    c. Oggetto;
    d. Temi trattati;

e. Decisioni prese;
f. Prossimi passi;
g. Prossimi incontri.

3) Questo verbale è gestito internamente dal team degli sviluppatori quindi il suo codice identificativo è a due livelli: il primo livello è diverso per ogni verbale relativo ad un incontro con il cliente e il secondo indica il numero di revisione interna;
4) Il verbale non è un oggetto versionabile, per cui è identificato univocamente dalla data dell'incontro con il cliente.

2.1.6. Documento di progetto

Descrizione

Il documento di progetto descrive i risultati ottenuti dall'analisi dei requisiti e fornisce l'idea che sarà alla base per la realizzazione di un sistema (software). Questo documento quindi oltre alla descrizione delle specifiche può contenere diagrammi dei casi d'uso che aiutano in modo schematico a comprendere quali sono le funzionalità messe a disposizione dal sistema.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_DocProgetto_versione.pdf (esempio: GDPRPrj_DocProgetto_v0.1.pdf);
2) Il documento viene scritto in linguaggio naturale strutturato senza la presenza di termini tecnici legati all'implementazione;
3) Deve contenere la data, l'autore del documento, il codice di versione ed eventualmente un glossario con i riferimenti a regolamenti o fonti esterne;
4) A differenza del documento: "Offerta in relazione al mandato", descritto al punto successivo, questo documento contiene una proposta di soluzione, ovvero descrive l'idea progettuale emersa dall'analisi dei requisiti ma senza fare un'offerta esplicita al cliente. Se l'idea di soluzione proposta in questo documento viene approvata dal cliente allora nell'incontro successivo verrà fatta un'offerta.

2.1.7. Offerta in relazione al mandato

Descrizione

Il documento con l'offerta al cliente è il naturale successore del documento di progetto; mentre il primo contiene le informazioni ottenute dall'analisi dei requisiti e quindi descrive quale sarà l'idea alla base della realizzazione di un sistema (software), il secondo contiene, oltre alle informazioni contenute nel documento di progetto, anche una proposta esplicita al cliente tenendo conto dei vincoli di tempo e dei costi.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_Offerta_versione.pdf (esempio: GDPRPrj_Offerta_v0.1.pdf);
2) L'offerta viene formulata in linguaggio naturale strutturato senza la presenza di termini tecnici legati all'implementazione;
3) Deve contenere la data dell'offerta, l'autore del documento, il codice di versione ed eventualmente un glossario con i riferimenti a regolamenti o fonti esterne;
4) Contiene una descrizione generale della soluzione proposta che descrive le funzionalità del sistema dal punto di vista dell'utilizzatore. Queste funzionalità dipendono dai requisiti forniti dal cliente e dalla loro successiva analisi svolta dal team di sviluppo.

2.1.8. Configuration Management

Descrizione

Il Configuration Management (abbreviato CM) è un documento indispensabile per stabilire e mantenere la coerenza delle prestazioni, funzionali e fisiche di un prodotto con i suoi requisiti, il design e le informazioni operative per tutta la sua durata. Nel caso specifico della realizzazione di un prodotto software gli obiettivi principali sono:

1) Identificare tutti gli item legati ad un certo prodotto software;
2) Tenere traccia di tutti i cambiamenti e del modo meno costoso per farli;
3) Garantire la consistenza fra il codice e le informazioni presenti nei documenti ad esso associati;
4) Tenere traccia dello stato del sistema e di tutte le sue possibili configurazioni;
5) Ridurre i rischi durante lo sviluppo e la manutenzione;
6) Aumentare la produttività diminuendo il numero di errori.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_CM_versione.pdf (esempio: GDPRPrj_CM_v0.1.pdf);
2) Il file contiene il numero di versione del CM, la data, il nome dell'autore della versione corrente, il nome del responsabile e le varie sezioni che costituiscono il corpo del documento.

2.1.9. Documento di avanzamento del progetto

Descrizione

Il documento di avanzamento del progetto contiene la descrizione dei progressi del team su un certo progetto. È una sorta di diario dei progressi suddiviso in date, dove per ogni data vengono riportati i passi compiuti (ad esempio consegne al committente, test eseguiti, eccetera) verso l'obiettivo finale.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà DocAvanzamento_data.pdf (esempio: DocAvanzamento_2018-11-23.pdf);
2) Il file contiene il numero di versione del documento, la data, il nome dell'autore della versione corrente, il nome del responsabile e il corpo del documento;
3) Il "corpo del documento" citato al punto precedente consiste di un elenco di date con descrizione dei passaggi portati a termine in quelle date ad esempio: "Consegnata al cliente versione 0.1 del documento di progetto";
4) Dal momento che le consegne al cliente corrispondono ad un incremento di 1 nel valore dell'indice di versione di secondo livello (o di primo livello se la consegna comprende nuove funzionalità), è fortemente consigliato indicare nel documento di avanzamento il numero di versione di un item ogni qual volta il suo indice di secondo livello (o di primo livello) viene incrementato.
5) Il documento non è versionabile, quindi è presente una singola versione che contiene tutti i progressi del team ordinati per data.

2.1.10. Ambito di lavoro

Descrizione

È stato scelto di mantenere un documento che descriva l'ambito di lavoro e che contenga informazioni sugli stakeholder relativi al progetto corrente.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_AmbitoLavoro_versione.pdf;
2) Il file deve essere diviso in due sezioni:
    a. Descrizione del dominio;
    b. Stakeholder.

2.1.11. Glossario

Descrizione

Il glossario, come suggerisce il termine, è un documento che contiene la descrizione per una serie di parole legate al dominio del progetto su cui il team sta lavorando che è bene siano note a tutti per una comunicazione più chiara con il cliente (che a differenza del team conosce bene il dominio).

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_Glossario_versione.pdf;
2) La struttura è molto semplice, è un documento con un elenco di parole legate al dominio del progetto in corso, ognuna accompagnata dalla sua descrizione.

2.1.12. Documento di test sui requisiti

Descrizione

Il documento di test sui requisiti è strettamente legato al documento dei requisiti, li analizza singolarmente e per ognuno di essi descrive un insieme di operazioni che deve essere possibile effettuare sul sistema per raggiungere i risultati descritti dal requisito. Quindi contiene un insieme minimo di test che vanno fatti per ogni requisito per verificare che le funzionalità descritte nel requisito siano effettivamente fornite dal sistema.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_TestSuRequisiti_versione.pdf;
2) Il file elenca i requisiti (distinti fra funzionali e non funzionali) e per ogni requisito presente nel documento dei requisiti è indicato un certo numero di azioni che il sistema deve mettere a disposizione.

2.1.13. Documento sulla base di dati

Descrizione

Il documento sulla base di dati contiene la descrizione delle tabelle che saranno presenti nel database del sistema in particolare: le chiavi primarie (attributi identificatori) di queste tabelle, le chiavi esterne (attributi che legano le diverse tabelle), le relazioni fra le tabelle, i vincoli sui valori di queste tabelle e il tipo di dato dei vari campi presenti nelle tabelle.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_DocBaseDati_versione.pdf;
2) Prima di descrivere in dettaglio le varie tabelle è presente una sezione che contiene una tabella con i nomi di tutte le tabelle del database e il loro prefisso che verrà ripetuto su ogni campo della tabella;
3) Per ogni tabella avremo un elenco dei suoi campi, il tipo di dato di ogni campo, un flag che indica se il campo è una chiave primaria o una chiave esterna e gli eventuali vincoli sui valori di quel campo;

4) Nel documento vanno descritte le relazioni che legano le tabelle (in linguaggio naturale oppure tramite un diagramma entità-relazione);
5) Ogni tabella deve avere una chiave primaria;
6) Per ogni chiave esterna va indicata la chiave primaria (e la sua tabella) a cui fa riferimento.

2.1.14. Documento di interazione

Descrizione

Il documento d'interazione descrive le modalità d'interazione fra l'utente finale ed il sistema descrivendo come può comportarsi l'utente nell'utilizzo delle funzionalità offerte dal sistema descritte nel documento di progetto. Per ognuna delle azioni messe a disposizione dal sistema (ad esempio per effettuare il login) vengono descritte le modalità d'interazione ovvero l'insieme dei passaggi che l'utente deve compiere per raggiungere il suo obiettivo (nel caso specifico effettuare il login) ed eventualmente alcuni ulteriori azioni correlate al raggiungimento dell'obiettivo (ad esempio se l'utente dimentica la password descrivo cosa può fare per recuperarla e poi effettuare il login).

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_DocInteraz_versione.pdf;
2) Il documento avrà un'introduzione che descrive sinteticamente il suo contenuto e poi una parte dedicata alle modalità d'interazione divisa in sezioni, ognuna delle quali descrive le modalità di interazione che hanno un certo obiettivo (ad esempio una sezione per il login/logout, una per interagire con il registro dei trattamenti, eccetera).

2.1.15. Test di accettazione

Descrizione

Il test di accettazione è strettamente legato al documento di test sui requisiti, è composto da una serie di tabelle che elencano i test da eseguire per ogni requisito (presi dal documento di test sui requisiti) e per ogni test è possibile barrare la casella corrispondente con un "sì" se il test ha successo oppure con un "no" se il test fallisce, inoltre è presente una casella libera in cui possono essere inserite eventuali note. Il documento contiene un insieme di test che vanno fatti per ogni requisito per verificare che le funzionalità descritte nel documento dei requisiti siano effettivamente fornite dal sistema. Il test di accettazione va svolto in presenza del cliente che al termine dell'incontro dovrà firmare il presente documento che quindi conterrà i risultati del test.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_TestAccettazione_versione.pdf;
2) Il file elenca i requisiti (distinti fra funzionali e non funzionali) e per ogni requisito presente nel documento dei requisiti è indicato un certo numero di test da effettuare sul sistema per dimostrare che rispetta tutti i requisiti.

2.1.16. Tempi di lavoro

Descrizione

Il team segna su un file xlsx i tempi di lavoro svolti sui vari progetti, in particolare è presente un file per ogni membro del team che si occuperà di tenerlo sempre aggiornato. In questo file viene descritto il contributo dei vari membri in termini di tempo sul progetto, questo servirà poi per avere una base su cui fare un'analisi dei costi.

Caratteristiche del file e contenuto minimo

1) Il nome del file sarà GDPRPrj_TempiLavoro_nome.xlsx;
2) Il file è una tabella Excel divisa in cinque colonne:
    a. Nome dello sviluppatore;
    b. Nome del progetto su cui ha lavorato (questo campo può assumere il valore "Interno" se il lavoro non è strettamente legato al progetto corrente ma più all'organizzazione del team ad esempio la scrittura del CM rientra in questo caso);
    c. Attività svolta (ad esempio: documentazione, test, CM, ricerca informazioni, eccetera);
    d. Data;
    e. Tempo di lavoro (espresso in minuti).

2.2. Change request

Una change request (abbreviato CR) è un documento che contiene una "chiamata" per una modifica di un sistema; è di grande importanza nel processo di gestione del cambiamento.

Le CR provengono in genere da una delle cinque fonti:

1) segnalazioni di problemi che identificano bug che devono essere corretti , che costituisce la fonte più comune;
2) richieste di miglioramento del sistema da parte degli utenti;
3) eventi nello sviluppo di altri sistemi;
4) cambiamenti nella struttura e negli standard sottostanti (ad esempio nello sviluppo del software questo potrebbe essere un nuovo sistema operativo);
5) richieste da parte dei dirigenti.

Ogni change request deve contenere: un codice univoco (numero della richiesta), autore della richiesta, data della richiesta, priorità, descrizione, moduli coinvolti (opzionale se esterna), tempo di lavoro stimato e successivamente data approvazione/rifiuto, data inizio modifiche.

Possono essere interne (esigenze degli sviluppatori) o esterne (esigenze del cliente); se sono esterne si pianifica la data di inizio delle modifiche (in base al tempo di lavoro stimato e alla priorità della modifica), se interne prima devono essere approvate (votazione a maggioranza fra gli sviluppatori).

Le change request possono essere presentate in formato cartaceo durante gli incontri (fra gli sviluppatori o con il cliente) oppure in formato digitale (ad esempio pdf) ma in entrambi i casi devono contenere le informazioni descritte in precedenza.

2.3. Codice

Il codice sorgente legato al progetto viene gestito tramite GitHub, ogni membro del team può leggerlo e modificarlo a condizione che gli altri membri siano d'accordo (se necessario va effettuata una change request per poter modificare il codice).

Ogni volta che uno degli sviluppatori effettua una modifica o aggiunge nuovo codice deve registrare i cambiamenti (per modifiche che introducono nuove funzionalità o risolvono bug è opportuno inserire un commento nel codice in cui si indica l'autore e la data della modifica e se presente il numero della change request associata alla modifica) e aggiornare il numero di versione del file modificato.

Ogni file contenente codice associato al progetto deve iniziare con qualche riga di commento che descrive il contenuto del file, inoltre il commento deve contenere la data di creazione del file e l'autore del file. In questo modo leggendo il commento sappiamo subito dove viene usato il file e a cosa serve, inoltre in caso di problemi è sempre possibile chiedere chiarimenti all'autore del file. Per i file di tipo HTML e CSS non è necessario effettuare questi passaggi, a meno che ad esempio un file HTML contenga al suo interno qualche script che sarebbe opportuno descrivere.

2.4. Gestione versioni

2.4.1. Criterio di numerazione

È stato scelto di numerare le versioni delle componenti del sistema nel seguente modo:

1) Ogni versione è identificata da un codice numerico a tre livelli (solo i primi due quando presentata al cliente);
2) Il terzo livello contiene il numero della revisione interna su un certo item, partendo da zero;
3) Il secondo livello contiene il numero della consegna al cliente (può essere zero solo in fase di sviluppo, quindi la prima consegna di un certo oggetto avrà 1 come codice di secondo livello);
4) Il codice di primo livello identifica un oggetto che mantiene obiettivi e caratteristiche fondamentali per tutte le sue consegne, questo codice è zero fino a che non si ottiene una versione completa, funzionante e utilizzabile dal cliente.
5) Esempio:
    a. 0.2 è la seconda consegna di un prototipo con alcune funzionalità minori ma che non può ancora essere considerato un sistema;
    b. 0.2.9 è la nona revisione interna sul prototipo sopra citato;
    c. 1.0.3 è la terza revisione interna della versione completa e funzionante del sistema non ancora rilasciata;
    d. 1.1 è la prima versione completa e funzionante del sistema consegnata al cliente;
    e. 3.1.15 è la quindicesima revisione interna della prima consegna della terza versione completa e funzionante del sistema.

2.4.2. Memorizzazione delle versioni

Per quanto riguarda la documentazione è stato deciso di mantenere nella repository GitHub del team tutte le versioni in formato pdf in modo che siano sempre disponibili per la consultazione mentre solo per l'ultima versione si è deciso di conservare anche il file in formato docx in modo che questo sia estensibile per passare alla versione successiva senza dover riscrivere tutto il documento. Verranno mantenute le versioni, non tutti i file associati a revisioni interne, quindi ad esempio file (con indice di terzo livello) come Nome Doc_v0.1.1 non saranno presenti nella repository a meno che questa non sia la versione su cui stiamo lavorando e quindi non ancora da consegnare al cliente.

2.5. Dati di test e relativa documentazione

2.5.1. Tipi di test e responsabili nominati

I possibili tipi di test che verranno eseguiti sul sistema sono i seguenti:

1) Black – box (funzionale);
2) White – box (strutturale);
3) Regression testing (prima di ogni consegna al committente);
4) Interface testing;
5) Stress testing;
6) Testing con tool di supporto (tool da definire);

7) Inspection.

Per ogni tipo di test è stato nominato un responsabile che si incarica di effettuare il testing seguendo le regole specificate successivamente e documentando in modo opportuno i risultati ottenuti. I responsabili per i diversi tipi di test sono:

- Baradel Luca: black – box, white – box;
- Di Benedetto Gianluca: regression testing, interface testing;
- Pellizzari Luca: stress testing, testing con tool di supporto, inspection.

2.5.2. Passi da seguire durante il testing

Per quanto riguarda i test di tipo dinamico (dal numero 1 al numero 6 compreso della sezione precedente) i passi da seguire sono i seguenti:

1) Specificare qual è il tipo di test che sta per essere effettuato;
2) Specificare data, autore e numero progressivo del test;
3) Elencare quali sono le componenti coinvolte;
4) Specificare quali sono gli input;
5) Specificare quali sono gli output previsti;
6) Specificare se ci sono ulteriori risorse utilizzate;
7) Descrivere i risultati ottenuti, quindi il comportamento del sistema (errore oppure comportamento corretto);
8) Aggiungere eventuali note.

I risultati del testing vanno riportati in un documento che deve contenere tutti i dati citati negli otto passaggi sopra descritti.

Per quanto riguarda il testing di tipo statico (inspection) questo può essere eseguito singolarmente oppure in gruppo. Se viene eseguito da una sola persona questa deve:

1) Registrare data, autore e numero progressivo del test;
2) Descrivere quali sono le componenti che verranno analizzate;
3) Descrivere in modo preciso quali sono i problemi riscontrati e dove si trovano.

Nel caso in cui questo processo venga eseguito in gruppo (da due o più persone) è possibile assegnare dei ruoli ai membri del gruppo: ad esempio una persona può leggere il codice, una può prendere nota dei problemi riscontrati, una può fare da “moderatore” dell’incontro, eccetera.

L’inspection è un processo che può essere utilizzato non solo per rilevare eventuali errori nel codice ma anche per controllare la correttezza di documenti come ad esempio UML, documenti di progetto, analisi dei requisiti, eccetera. In questi casi i passi da seguire sono gli stessi rispetto a quelli descritti per l’inspection del codice: registrare data, autore e numero progressivo dell’inspection, identificare l’elemento che si vuole analizzare e registrare i problemi riscontrati.

2.5.3. Criteri di nominazione per test e risultati

Un criterio di nominazione per i test può essere il seguente:

1) GDPRPrj_CT_Rxy_v1.1 per il file che riguarda la test chain relativa alla versione 1.1 del requisito xy. La struttura del file può essere la seguente: setting, dataset, passi/risultato;

2) GDPRPrj_RT_Rxy_18122_v1.1 per il file che conterrà i risultati del test numero 18122 sulla versione 1.1 del requisito xy. La struttura del file può essere la seguente: data test, esecuzione test, risultato finale, note (ad esempio indicazioni di qualcosa che non è stato considerato nel test).

2.6. Database

2.6.1. Criteri di nominazione

È stato deciso di inserire nelle pagine iniziali del documento sulla base di dati una tabella che descrive i criteri che verranno utilizzati nel progetto corrente per assegnare i nomi alle tabelle della base di dati e ai campi contenuti in esse. I nomi delle tabelle saranno scritti in lingua inglese e i nomi dei campi delle tabelle saranno preceduti da un prefisso che dipende dal nome della tabella a cui appartengono. Un esempio potrebbe essere:

1) nomi delle tabelle: events, subjects;
2) nomi dei campi appartenenti alla tabella events: e_id, e_date, e_type, eccetera;
3) nomi dei campi appartenenti alla tabella subjects: s_id, s_name, s_surname, eccetera.

Quindi per la tabella events abbiamo che ognuno dei suoi campi sarà nominato nel seguente modo: “e” + “_” + “nome del campo”, mentre per la tabella subjects abbiamo: “s” + “_” + “nome del campo”. I prefissi scelti in questo caso sono “e” ed “s”, nel caso si abbiano più tabelle il cui nome inizia con la stessa lettera il team sceglierà un nuovo prefisso in modo che sia univoco per ogni tabella.

Questi prefissi verranno riportati nel documento sulla base di dati e in linea di massima saranno diversi per ogni progetto in quanto dipendono strettamente dal nome delle tabelle (che viene scelto in base ai dati che una tabella contiene).

2.6.2. Gestione file legati alla base di dati

Dopo aver creato la base di dati e aver inserito le tabelle con i relativi vincoli è necessario avere una copia delle query di creazione delle tabelle e dei vincoli per poter ricreare la base di dati su un altro calcolatore oppure semplicemente per poter ricostruire velocemente le tabelle nel caso in cui vengano introdotti degli errori o vengano cancellate erroneamente alcune tabelle (o alcuni vincoli). Per questo verrà mantenuto un file in formato .sql che contiene le query di creazione delle tabelle e delle loro relazioni (e vincoli). Questo file sarà versionato (oppure datato) in modo da sapere su quale versione stiamo lavorando nel caso in cui ce ne siano più di una. Oltre a questo file sarà presente un data dictionary relativo alla base di dati che contiene informazioni relative alle varie tabelle, alle loro relazioni, ai campi presenti in ogni tabella e agli indici eventualmente associati ai campi. Anche il data dictionary sarà versionato (o datato) perché è strettamente legato alle informazioni di una specifica base di dati. È possibile che entrambi i file sopra descritti (data dictionary e query sql) siano generati automaticamente dal tool utilizzato per gestire la base di dati.

2.7. Software di supporto

2.7.1. GitHub

Uno dei tool di supporto utilizzati dal team GL³ è GitHub, usato come repository per tutta la documentazione collegata al progetto. Il responsabile di questo item deve assicurarsi che la repository contenga tutti e soli i file indicati nel CM e che tutti questi file abbiano la struttura stabilita dal team e descritta nel CM.

Il team inoltre utilizza la wiki messa a disposizione da GitHub per pianificare il lavoro da svolgere. Ogni membro del team ha una sua pagina wiki con la lista dei “to do” (compiti da svolgere) eventualmente

accompagnati da una scadenza e da una descrizione molto sintetica. In questo modo ogni membro del team può gestire il lavoro che gli spetta e tenere sotto controllo lo stato di avanzamento per i vari compiti da svolgere.

2.7.2. Xampp

Xampp è il tool scelto dal team GL3 per gestire la base di dati. Il tool è utilizzabile offline e mette a disposizione un'interfaccia molto semplice per la gestione delle varie tabelle di una base di dati; inoltre permette di generare in modo automatico i due file di cui si parla nella sezione "2.6.2 Gestione file legati alla base di dati" del presente documento.