



Utilizzo del server per C04/PQS

Versione 3.0

Stato del documento:

Formale ed

Interno

Sommario :

Questo documento definisce istruzioni e norme per l'utilizzo del server a supporto del capitolato C04: "Sistema *software* per l'informatizzazione della gestione di qualità a norma ISO 9000:2000 nelle scuole".

Redazione

| Nominativo | Ruolo | Data |
|----------------|----------------|------------------|
| Roberto Pordon | Amministratore | 20 gennaio 2007 |
| Eric Miotto | Amministratore | 19 febbraio 2007 |

Lista di Distribuzione

| Nominativo | Ruolo |
|-----------------------|------------------------------|
| Tullio Vardanega | Committente |
| Renato Conte | Committente |
| Lucia Meneghello | Verificatore |
| Margherita Collicelli | Responsabile |
| Eric Miotto | Amministratore/Programmatore |
| Stefano Gazzola | Programmatore |
| Roberto Pordon | Verificatore |
| Lorenzo Daniele | Programmatore |
| Alberto Meneghello | Programmatore |

Approvato da:

| Versione | Nominativo | Data |
|----------|-----------------------|---------------|
| 3.0 | Margherita Collicelli | 16 marzo 2007 |

Registro delle Modifiche:

| Versione | Autore | Data |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|------------------|
| 3.0 | Margherita Collicelli | 16 marzo 2007 |
| Approvazione documento per Revisione di Qualifica. | | |
| 2.3 | Lucia Meneghello | 16 marzo 2007 |
| Effettuata verifica del documento e rigenerato l'indice. | | |
| 2.2 | Eric Miotto | 02 marzo 2007 |
| Integrata sezione 4.2. | | |
| 2.1 | Eric Miotto | 27 febbraio 2007 |
| Inserita sezione sul formato dei messaggi in svn (sezione 4.2). Aggiornate regole di commento modifiche in Trac (sezione 5.2) e rivista sezione sui ticket (sezione 5.4). | | |
| 2.0 | Lorenzo Daniele | 23 febbraio 2007 |
| Approvazione documento per Revisione di Progetto Definitiva. | | |
| 1.2 | Lucia Meneghello | 22 febbraio 2007 |
| Effettuata verifica del documento. In particolare corretta la forma di alcune frasi. | | |
| 1.1.1 | Lucia Meneghello | 21 febbraio 2007 |
| Aggiornata lista di distribuzione. | | |
| 1.1 | Eric Miotto | 19 febbraio 2007 |
| Aggiornata lista di distribuzione. Aggiornata sezione 4.1 (sconsigliato l'uso di lettere accentate nei nomi dei file), inserite sezioni 4.3 e 4.7 sull'uso dei lock e di Netbeans con Subversion. | | |
| 1.01 | Roberto Pordon | 11 febbraio 2007 |
| Aggiunte alcune norme per l'utilizzo di Svn, BoUML, e Trac | | |
| 1.0 | Roberto Pordon | 02 febbraio 2007 |
| Approvazione documento per Revisione di Progetto Preliminare | | |
| 0.4 | Lorenzo Daniele, Alberto Meneghello | 01 febbraio 2007 |
| Verificato il contenuto del documento. Corretti lievi errori sintattici. | | |
| 0.3 | Roberto Pordon | 30 gennaio 2007 |
| Aggiunte regole per Subversion e informazioni su Trac | | |
| 0.2 | Roberto Pordon | 26 gennaio 2007 |
| Aggiunte informazioni per la risoluzione di conflitti e uso del Repository Browser, aggiunte note di utilizzo, corretti alcuni errori | | |
| 0.1 | Roberto Pordon | 20 gennaio 2007 |

Prima stesura del documento.

Indice

| | | |
|----------|--------------------------------------------------------------|-----------|
| 1 | Introduzione..... | 6 |
| 1.1 | Scopo del documento..... | 6 |
| 1.2 | Riferimenti..... | 6 |
| 2 | Descrizione del server..... | 7 |
| 2.1 | Servizi offerti..... | 7 |
| 2.2 | Indirizzo..... | 7 |
| 2.3 | Accesso..... | 7 |
| 3 | Guida all'utilizzo di Subversion con TortoiseSVN..... | 8 |
| 3.1 | Che cos'è TortoiseSVN..... | 8 |
| 3.2 | Installazione..... | 8 |
| 3.3 | Creare una directory di lavoro (working copy)..... | 8 |
| 3.4 | Aggiornamento della working copy..... | 9 |
| 3.5 | Modificare i file..... | 9 |
| 3.5.1 | <i>Aggiungere un file o una directory.....</i> | <i>10</i> |
| 3.5.2 | <i>Rinominare.....</i> | <i>10</i> |
| 3.5.3 | <i>Spostare.....</i> | <i>10</i> |
| 3.5.4 | <i>Eliminare.....</i> | <i>10</i> |
| 3.5.5 | <i>Nota importante.....</i> | <i>10</i> |
| 3.6 | Caricare le modifiche..... | 10 |
| 3.7 | Risoluzione di conflitti..... | 11 |
| 3.8 | I nostri repository..... | 11 |
| 3.9 | Utilizzo senza working copy..... | 12 |
| 3.10 | Nota finale..... | 12 |
| 4 | Norme per l'utilizzo di Subversion..... | 13 |
| 4.1 | Nomi di file..... | 13 |
| 4.2 | Formato messaggi..... | 13 |
| 4.3 | Uso dei lock..... | 14 |
| 4.4 | Modifiche al file system del repository..... | 14 |
| 4.5 | Regole importanti..... | 14 |
| 4.6 | Caricamento di file BoUML..... | 14 |
| 4.7 | Uso con Netbeans..... | 14 |
| 5 | Utilizzo di Trac..... | 15 |
| 5.1 | Ambienti..... | 15 |
| 5.2 | Modifica dell'ambiente..... | 15 |
| 5.3 | Wiki..... | 15 |
| 5.3.1 | <i>Nomi delle pagine.....</i> | <i>15</i> |
| 5.3.2 | <i>Pagine Personali.....</i> | <i>15</i> |
| 5.4 | Ticket..... | 16 |
| 5.4.1 | <i>Creazione di ticket.....</i> | <i>16</i> |
| 5.4.2 | <i>Chiusura e riapertura di ticket.....</i> | <i>16</i> |

1 Introduzione

1.1 Scopo del documento

Questo documento intende fornire le istruzioni di base per utilizzare i servizi offerti dal server e successivamente definisce le principali norme per l'utilizzo dei servizi stessi.

1.2 Riferimenti

[SVNB] Subversion Book, versione 1.2, <http://svnbook.red-bean.com/>

2 Descrizione del server

2.1 Servizi offerti

Il server offre i seguenti strumenti:

- server web: **Apache2**;
- Version Control System: **Subversion (Svn)**;
- Issue Tracking System: **Trac**.

2.2 Indirizzo

Il server si trova all'indirizzo <https://egoless.mine.nu>.

2.3 Accesso

L'accesso al server è consentito solo agli utenti autenticati. Ogni membro del gruppo è invitato a fornire agli amministratori una coppia username-password per l'accesso ai servizi del server. Lo username e la password sono a libera scelta. La password dovrà essere il più sicura possibile (cioè contenere lettere sia maiuscole che minuscole e magari qualche numero o carattere speciale).

Ogni utente è tenuto a mantenere segreta la propria password e a fare in modo che non possa essere scoperta da altri.

3 Guida all'utilizzo di Subversion con TortoiseSVN

3.1 Che cos'è TortoiseSVN

TortoiseSVN è un'interfaccia che si integra con Windows e consente di lavorare in maniera semplice e rapida con Subversion, il sistema di controllo versione messo a disposizione sul server.

3.2 Installazione

Scaricate TortoiseSVN (<http://tortoisesvn.tigris.org>) e installatelo sul vostro pc (richiede riavvio del sistema operativo).

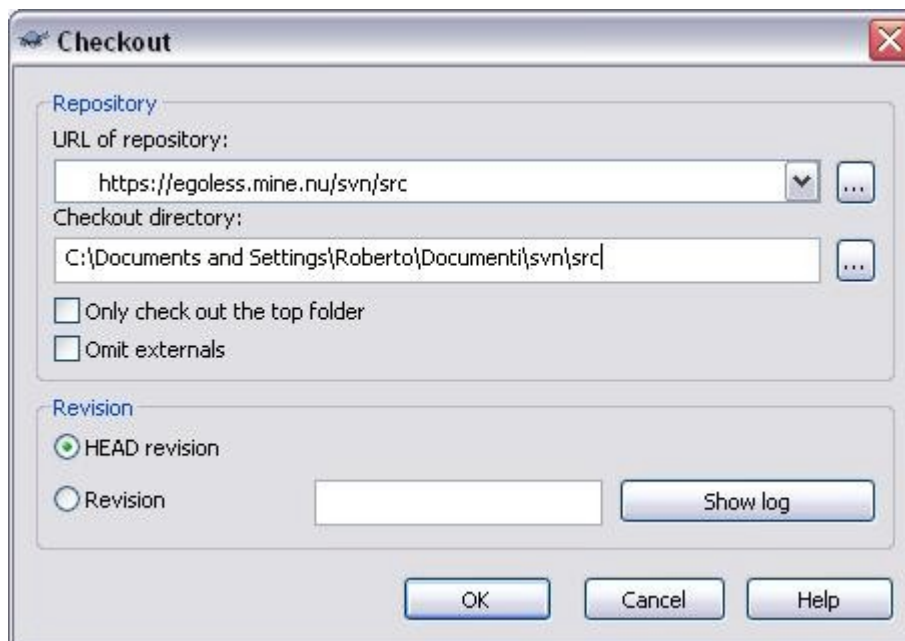
3.3 Creare una directory di lavoro (working copy)

Per poter lavorare è necessario creare una copia in locale (sul nostro pc) del repository. Questa copia è chiamata “working copy”. Per poter fare qualsiasi modifica al repository è necessario lavorare in una working copy e poi trasferire il contenuto sul server tramite un apposito comando. Ciò significa, in parole povere, che se ad esempio voglio aggiungere un file al repository non posso caricarlo direttamente (come in Google Groups) ma devo *prima* aggiungerlo alla mia working copy locale e *poi* trasferire il tutto sul server. Non preoccupatevi, non è difficile come sembra.

Create una nuova directory (dove preferite) e fate clic col tasto destro sull'icona. Se TortoiseSVN è stato installato correttamente dovrebbe comparire l'opzione “SVN Checkout...”. Fate clic e comparirà una finestra di dialogo che richiederà l'inserimento dell'URL del repository.

I nostri repository si trovano all'URL: https://egoless.mine.nu/svn/nome_repository

Questa è la finestra che dovrebbe comparire e che crea una working copy del repository “src”.



Vi verrà richiesto di autenticarvi. Se l'operazione va a buon fine TortoiseSVN dovrebbe scaricare tutto il contenuto del repository nella vostra working copy.

La procedura è ovviamente identica per gli altri repository. Create una working copy per ogni repository su cui volete lavorare.

3.4 Aggiornamento della working copy

Prima di cominciare a lavorare sui file è importante aggiornare la vostra working copy in modo da avere tutti i file più recenti a disposizione. Naturalmente si può lavorare anche senza fare prima l'aggiornamento, ma è sconsigliato perché è molto più probabile che vi ritroviate a dover integrare le vostre modifiche con quelle degli altri. Come regola quindi *fate sempre l'aggiornamento* prima di iniziare a modificare i file.

Per aggiornare è sufficiente un clic col tasto destro sulla directory contenente la working copy. Scegliete la voce "Update" nel menù "TortoiseSVN" e l'aggiornamento avrà inizio. Comparirà una finestra che mostrerà i file aggiornati. Una volta completato l'aggiornamento fate clic su ok e potrete iniziare le modifiche.

3.5 Modificare i file

La modifica del contenuto dei file non richiede operazioni particolari. Usate il vostro editor come al solito.

Le operazioni che richiedono un po' di attenzione sono quelle di creazione, ridenominazione, spostamento ed eliminazione dei file. Non è possibile infatti effettuare queste operazioni come per qualsiasi file del disco, ma bisogna utilizzare

l'interfaccia di TortoiseSVN.

3.5.1 Aggiungere un file o una directory

Create normalmente il file (o la directory) nella working copy e fate clic col tasto destro. Scegliete quindi la voce "Add" nel solito menù "TortoiseSVN".

3.5.2 Rinominare

Per rinominare un file è necessario usare il Repository Browser. Fate clic col tasto destro sulla directory contenente la working copy e scegliete "Repo-browser" nel menù "TortoiseSVN". Comparirà una finestra che vi mostrerà il contenuto del repository come struttura ad albero. Facendo clic con il tasto destro sui file potrete rinominarli scegliendo l'opzione "Rename". Comparirà una finestra di dialogo in cui potrete inserire un commento all'operazione.

3.5.3 Spostare

Anche per spostare un file usate il Repository Browser e spostate i file con il classico "drag-and-drop". Potrete inserire un commento allo spostamento.

3.5.4 Eliminare

Per eliminare un file, oltre ad usare il solito Repository Browser, potete scegliere dalla working copy l'opzione "Delete" nel menù "TortoiseSVN".

3.5.5 Nota importante

Le operazioni di ridenominazione, spostamento ed eliminazione dei file possono portare ad inconsistenze nel repository e a duplicazioni dei file. Per evitare problemi effettuate queste operazioni solo se strettamente necessario e magari comunicatele preventivamente agli altri membri del gruppo.

Se usate il Repository Browser per le modifiche ricordatevi poi di aggiornare la vostra working copy.

3.6 Caricare le modifiche

Il comando "Commit" nel solito menù vi permetterà di caricare sul server le vostre modifiche ai file. Se nel frattempo non ci sono state modifiche da parte di altri il comando "Commit" si concluderà senza problemi. Se invece qualcuno ha modificato i file contemporaneamente a voi, TortoiseSVN si rifiuterà di completare l'operazione e richiederà di aggiornare la working copy. L'aggiornamento risulterà in un altro errore: conflitto tra i file.

3.7 Risoluzione di conflitti

Le icone dei file in conflitto saranno segnate da un triangolo giallo come in questo esempio



In caso di conflitto SVN creerà nella working copy dei file utili per gestire la situazione. Come esempio poniamo di avere un conflitto del file `murphy.cpp`. Nella working copy compariranno tre nuovi file:

- **`murphy.cpp.mine`**: questo file contiene la vostra copia più aggiornata del file (cioè quella di cui avete tentato di fare il commit).
- **`murphy.cpp.rX` e `murphy.cpp.rY`**: dove X e Y sono interi con $Y > X$. Il file di estensione `.rX` contiene l'ultima versione del file che avevate scaricato prima di iniziare le modifiche. Il file di estensione `.rY` contiene la versione che è stata caricata sul server dopo che voi avete iniziato a lavorare.

Il file `murphy.cpp` (senza ulteriori estensioni) conterrà dei marcatori speciali (inseriti da SVN) nelle righe in conflitto.

Per risolvere il conflitto ci sono due metodi suggeriti (non sono gli unici. Per trovarne altri consultate la guida di SVN).

La prima possibilità è di usare (dal solito menù) il comando "Edit conflicts". Si aprirà l'interfaccia "TortoiseMerge" che vi permetterà di confrontare le righe in conflitto e di decidere come risolvere la situazione.

Questo metodo funziona bene per file in formato testo (ad es. i file sorgente) ma è scomodo, se non inutilizzabile, per file binari (come i documenti di OpenOffice).

In questo caso il consiglio è quello di "fare a mano". Potete cioè modificare come volete i vostri file. L'importante è che al termine delle modifiche sia presente nella working copy il file `murphy.cpp` e che questo contenga la versione corretta. A questo punto scegliete l'opzione "Resolved..." che cancella i file temporanei e riprovate il comando Commit.

3.8 I nostri repository

Questa è la lista dei repository disponibili sul server. Ricordo che si trovano all'indirizzo https://egoless.mine.nu/svn/nome_repository :

- **`sand`**: repository per prove e esperimenti per capire come funziona Svn;
- **`src`**: codice sorgente dell'applicazione;

- **doc:** documenti del progetto e presentazioni;
- **examples:** i progetti GAAS e SGPEM;
- **ref:** documenti di consultazione;
- **mmedia:** immagini, registrazioni, video utili per il progetto.

3.9 Utilizzo senza working copy

I repository `examples`, `ref`, e `mmedia` non contengono (in linea generale) file soggetti a revisione, ma solo materiale utile al progetto. Se non volete creare una working copy per questi repository potete farlo. Il contenuto dei repository è in ogni caso visualizzabile (ma non modificabile) con un qualsiasi browser. Inserendo l'indirizzo del repository comparirà semplicemente una pagina con la lista dei file.

Per modificare un repository senza farne una working copy usate il Repository Browser di TortoiseSVN (che potete aprire in ogni momento sul vostro Desktop). Potete modificare i file esistenti (come già spiegato sopra) e anche caricare nuovi file usando il menù che compare facendo clic col tasto destro sull'icona della directory radice del repository.

3.10 Nota finale

Le informazioni qui contenute sono quelle basilari per iniziare velocemente a lavorare. Il consiglio è, se possibile, di dare uno sguardo anche alla guida di TortoiseSVN e di Subversion per capirne il funzionamento più in dettaglio e, soprattutto, in maniera più chiara. Il modo più semplice per imparare resta comunque quello di fare delle prove. A questo scopo è stato creato un apposito repository all'indirizzo <https://egoless.mine.nu/svn/sand> dove potrete fare tutti gli esperimenti che volete. Buon lavoro!

4 Norme per l'utilizzo di Subversion

4.1 Nomi di file

Per evitare problemi con file systems che non supportano nomi *case-sensitive*, non dovranno comparire nella stessa directory file il cui nome differisca soltanto per il *case*.

Per evitare problemi di compatibilità, è sconsigliato inserire lettere accentate nei nomi.

Il nome dei file inoltre non dovrà contenere il numero di versione del file (come invece accadeva su Google Groups).

4.2 Formato messaggi

Quando si fa commit o si acquisisce un lock (vedi sezione 4.3) è obbligatorio specificare un messaggio che spieghi rispettivamente le modifiche apportate al repository o le ragioni per cui si è acquisito il lock.

All'interno dei commenti non usare lettere accentate, perché non tutti i client le supportano.

Si supponga di aver modificato n documenti. Il formato di messaggio per il commit è il seguente:

```
* [Nome documento 1] [Versione documento 1]: [spiegazione modifica al
documento 1]
* [Nome documento 2] [Versione documento 2]: [spiegazione modifica al
documento 2]
...
* [Nome documento n] [Versione documento n]: [spiegazione modifica al
documento n]
```

Alcuni commenti:

- quando si fa un commit in Subversion, si crea una nuova versione dell'intero repository. Il messaggio è dunque riferito all'intero repository e deve specificare quali file sono stati modificati;
- siccome il messaggio può essere visualizzato tramite Trac, si approfitta della formattazione wiki per la lista (ricordarsi a tal proposito dello spazio prima dell'asterisco);
- la versione del documento è quella scritta dell'ultima riga del registro delle modifiche, non quella del repository.

Lo stesso formato si applica anche per la modifica di progetti. Le modifiche vanno

separate per progetto e non per singolo file sorgente modificato.

4.3 Uso dei lock

Per alcune tipologie di file la filosofia copy-modify-merge adottata da Subversion non è adatta: si pensi alle immagini o ai documenti fatti in OpenOffice. Per modificare questi tipi di file, si deve prima acquisire un lock, poi si può procedere alla modifica.

Il lock deve essere trattenuto per il tempo strettamente necessario all'esecuzione della modifica, poi va eseguito un unlock.

E' **severamente** proibito rompere e/o rubare il lock (breaking and stealing the lock) che un utente ha su un determinato file.

4.4 Modifiche al file system del repository

Come già spiegato, le modifiche al file system di un qualsiasi repository, dovranno essere solo quelle strettamente necessarie e possibilmente dovranno essere esplicitamente comunicate agli altri membri del gruppo.

4.5 Regole importanti

1. Fare sempre l'aggiornamento della propria working copy prima di iniziare le modifiche.
2. Inserire sempre una descrizione delle modifiche effettuate prima di completare un *commit* o prima di acquisire un lock.

4.6 Caricamento di file BoUML

I diagrammi UML creati da BoUML sono salvati non in un unico file (come ArgoUML) ma in una cartella. Per caricare su svn i diagrammi creati con BoUML utilizzate questa procedura:

- create il diagramma normalmente, salvandolo sotto la cartella UML;
- quando avete finito di fare le azioni, chiudere BoUML. In questo modo sparisce un file di lock che non deve *assolutamente* essere messo nel repository;
- a questo punto aggiungete la cartella del diagramma al repository e fate commit;
- poi potete riaprire il file ed aggiungere diagrammi;
- quando aggiungete un diagramma, assicuratevi prima del commit, di aggiungere il file che corrisponde al diagramma creato (ve ne accorgete perché manca l'icona di TortoiseSVN o perché "svn status" vi dà sul file un punto di domanda).

4.7 Uso con Netbeans

Il supporto per Subversion è integrato direttamente in Netbeans, per cui non è necessario creare prima la working copy per il progetto. E' sufficiente creare il progetto in un directory e poi importarlo in svn da Netbeans, oppure fare checkout di un progetto esistente in svn direttamente da Netbeans. Netbeans si occuperà in tutti i casi di invocare i giusti comandi e di rendere la directory del progetto una working copy.

5 Utilizzo di Trac

5.1 Ambienti

Sono a disposizione due istanze di Trac:

- https://egoless.mine.nu/PQS_src (per il repository src);
- https://egoless.mine.nu/PQS_doc (per il repository doc).

5.2 Modifica dell'ambiente

Ogni modifica ad una pagina wiki deve essere commentata utilizzando la casella di testo "Comment about this change".

Ogni modifica ad un ticket deve essere commentato nella casella di testo "Comment", a meno che:

- non sia stato cambiato il tipo;
- non sia stato chiuso.

5.3 Wiki

5.3.1 Nomi delle pagine

I nomi delle nuove pagine dovranno essere scelti secondo il formato CamelCase indicato nella pagina https://egoless.mine.nu/PQS_doc/wiki/WikiPageNames

5.3.2 Pagine Personali

Per ogni componente del gruppo è stata creata una pagina personale nella wiki di PQS_doc. L'indirizzo è https://egoless.mine.nu/PQS_doc/wiki/NomeCognome.

Questa pagina serve a contenere le informazioni sulla persona, i contatti, e le ore dedicate al progetto. Ogni componente deve annotare nella sezione "Ore di lavoro" il tempo dedicato al progetto in questa forma:

data, ore dedicate: lavoro svolto

Si inseriscano le righe in modo che la prima sia la più recente. Esempio:

24 gennaio 2007, 2h 30m: Redazione Specifica Tecnica

19 gennaio 2007, 1h 45m: Aggiornamento Analisi dei requisiti

18 gennaio 2007, 3h: Riunione con Swell Systems

Si consiglia di aggiornare frequentemente la propria pagina personale così da poter avere in ogni momento la situazione aggiornata sull'impegno di ogni componente.

5.4 Ticket

5.4.1 Creazione di ticket

I ticket servono per due scopi:

- pianificare le attività da svolgere;
- segnalare un problema riscontrato sui documenti o sul codice in fase di verifica.

I ticket per la pianificazione devono avere tipo task o enhancement e devono essere creati ed assegnati dal responsabile.

I ticket per la verifica devono avere tipo defect e devono essere creati da un verificatore. Il ticket deve essere assegnato alla persona responsabile del codice o del documento in cui è stato rilevato il problema

5.4.2 Chiusura e riapertura di ticket

Il possessore di un ticket è tenuto a chiuderlo appena considera il problema risolto ed ha reso disponibili i risultati della soluzione (commit del documento o del codice). Il ticket potrà essere riaperto se il responsabile o il verificatore ritiene che l'attività associata non è stata ancora completata o non ha avuto l'effetto sperato.