

Università degli Studi di Genova
Scuola Politecnica
Corso di Laurea Magistrale in Ingegneria Informatica

PROGETTO RELATIVO AL CORSO DI
SOFTWARE ENGINEERING
(PROGETTO 31- **GESTIONE LABORATORI VIRTUALI**)

STUDENTE:

ROSSI RICCARDO

DOCENTE:

PROF. NARIZZANO MASSIMO

CLIENTE:

ISTITUTO NAUTICO SAN GIORGIO



Contesto

2

Gestione calcolatori di un ambiente scolastico affetto da una carenza cronica di **risorse** e **assistenza tecnica**.

Problema

- Utilizzo del calcolatore diventato fondamentale nelle scuole, in particolare in Istituti Tecnici.
- Aule di laboratorio piene di computer datati con risorse fisiche scarse.
- Calcolatori vincolati a software obsoleto.
- Software vincolato ad un determinato sistema operativo (più sistemi operativi nello stesso pc).
- Enorme quantità di tempo impiegato dai tecnici di laboratorio per «predisporre» un'aula.
- Fondi a disposizione delle scuole molto scarsi e non sufficienti a fornire una buona dotazione tecnologica.

Obiettivo

Software per la gestione di un ambiente scolastico mediante infrastruttura di calcolo **virtualizzata**.

- Unico server in grado di gestire tutti i calcolatori di un intero ambiente scolastico (e non solo!)

GLV

- Utilizzo della piattaforma **oVirt** (alternativa *open-source* che più si ispira alla soluzione commerciale *VMware vSphere*).
- Creazione nuova macchina virtuale **personalizzata**.
- Pool di macchine virtuali già esistenti.
- Gestione macchine virtuali sul client (avvio, arresto, visualizzazione, eliminazione).

Vincoli

Installazione sul client di **Remote-Viewer** (molto «leggero») per la visualizzazione della macchina virtuale.

Approccio iniziale

- Identificazione del problema (quali sono le esigenze del cliente ?).
- Identificazione delle funzionalità del software.
- Scrittura del *User Requirements Document* (traduzione delle funzionalità in **requisiti**).
- Analisi del documento da parte del cliente (rispecchia le necessità? I requisiti sono quelli richiesti?).

Dai requisiti...

- Documentarsi (di cosa ho bisogno per lo sviluppo?).
- Ideare possibili soluzioni per implementare le funzionalità del software.
- Disegnare grafici e diagrammi.
- Stesura del documento di progettazione (*DRS*).

Codice

5

HTML/CSS, JavaScript (con jQuery), PHP.

Utilizzo *di AJAX* per eseguire le chiamate alle *API* di *oVirt*.

- Installazione ambiente di lavoro (configurazione rete locale, installazione ***oVirt Engine*** ed ***oVirt Node*** con relative configurazioni, caricamento *ISO* dei sistemi operativi etc...).
- Scrittura del codice (è importante mantenersi il più possibile fedeli al *DRS*, tuttavia è stata necessaria qualche modifica e aggiunta durante la scrittura).

Testing

6

Test Strutturale:

- Trovare dati di test che consentono di percorrere tutto il programma.
- Utilizzo di *JSCoverage* per la copertura del codice.

Test Funzionale:

- Simulare l'utente che utilizzerà il software al fine di identificare eventuali errori e anomalie nel software.

Documentazione

- Scrittura del file *README.md* per «spiegare» a chi utilizzerà GLV come installare le funzionalità ed avviare il software.
- Documentazione ottenuta commentando il codice attraverso caratteri specifici (generata con *JSDoc*).

Software

7

Dimostrazione pratica del software.

Problemi riscontrati

- Difficile comprensione iniziale riguardo le specifiche del software.
- Documentazione *oVirt* quasi inesistente e sito web ufficiale molto confusionario e con contenuti obsoleti e poco chiari.
- Pochissimo materiale sul web e difficoltà a trovare utenti che utilizzano la soluzione.
- Configurazione dell'ambiente di lavoro molto complessa.

Futuro di GLV

- Proseguimento come **tesi di laurea**.
- Implementazione GLV su larga scala.
- Utilizzo di GLV per gestire le *LIM* con relativi driver.
- Nuove specifiche ancora da decidere.

Pro e Contro del corso

Pro:

- Sviluppo di nuove conoscenze.
- Prima esperienza vera e propria come ingegnere.
- Uso di GitHub.
- Contatto con persone esterne all'università.

Contro:

- Grandissima quantità di tempo per completare tutti i passi del progetto.
- Necessità di troppe conoscenze preliminari per lo sviluppo.

***GRAZIE A TUTTI
PER L'ATTENZIONE***

Rossi Riccardo

Red.riccardo.91@gmail.com