

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

Progetto relativo al corso di *Software Engineering* (Progetto 46)

Studente:

De Luca Jacopo

Docente:

Prof. Narizzano Massimo

Azienda:

WeSii s.r.l.



Contesto

Geolocalizzazione dei malfunzionamenti nell'impianto fotovoltaico con l'obiettivo di aumentare la produzione del cliente.

- Raccolta dati con droni
- **Elaborazione immagini**
- Interfaccia con l'utente

Problema

Numero di pannelli da analizzare (migliaia di moduli!).

Molti passaggi automatizzati, altri necessitano ancora di **interventi manuali.**

Obiettivo

Automatizzazione del processo di **riconoscimento** di ogni singolo pannello solare all'interno delle immagini del drone.

Tool di uso interno a WeSii.

Solar Panel Analyzer (SPA)

- Immagini input (drone)
- Elaborazione automatica
- Raccolta dati dei pannelli
- Immagini pannelli e informazioni pannelli come output
- Interfaccia grafica essenziale

Di cosa ha bisogno il cliente?

- Identificazione del problema (esigenze del cliente)
- Identificazione delle funzionalità del software
- Stesura dell'**URD** (User Requirements Document)
- Analisi del documento scritto (rispecchia le necessità?)

Fondamentali:

- la **comprensione** delle esigenze del cliente (priorità)
- la **traduzione** delle esigenze **in requisiti** del software

Come?

Dai requisiti identificati alla **progettazione** del software:

- Analizzare l'URD
- Documentarsi (Quali strumenti servono? Come usarli?)
- Ideare possibili soluzioni per l'implementazione delle funzionalità
- Disegnare grafici
- Stesura **DRS** (Design Requirement Specification Document)

Nel DRS:

- Architecture
- Use cases
- Interfacce
- Structural design (Class and Object diagram)
- Data stores
- Dynamic model

Il codice

Python 3, OpenCV, altre librerie (wxPython, tiff file, etc.)

- Installazione dell'ambiente di lavoro.
- Mantenersi fedeli al DRS, anche se durante la scrittura del codice è stata necessaria qualche modifica/aggiunta.
- Algoritmo di riconoscimento (procedura basata sul **riconoscimento della geometria** nelle immagini).

Risolvere problemi in fase di coding:

- Compatibilità
- Uso della memoria

Testing

Test strutturale:

- Identificare il percorso che utilizza tutte le funzionalità del tool
- coverage.py per calcolare la copertura (94% di Branch Coverage)
- unittest.py per testare il modulo Elaboration.py (**codice**)

Test funzionale:

- Simulare l'utente nell'utilizzo del software

Documentazione

- Doxygen per generare documentazione direttamente dal codice commentato attraverso #
- Stesura file README.md per spiegazioni (installazione, running, testing)

Il software

Dimostrazione pratica del software.

Problemi incontrati

- Installazione librerie in versione Ubuntu non recente (Xubuntu)
- Uso di strumenti mai incontrati (Python e OpenCV)
- Identificazione dei parametri
- Elaborazione delle immagini .TIF (Out of memory! Molto grandi)
- Riconoscimento dello stesso pannello in due immagini diverse
- Aggiornamento librerie wxPython (GUI)
- Testing: poche informazioni su testing in Python
- Documentazione: tool (tipo javadoc) poco pratici e poche informazioni
- Mantenere il software fedele a URD e DRS

PRO

- **Prima esperienza come Ingegnere Informatico nella pratica**
- **Sviluppo di nuove conoscenze (documentazione, strumenti e problemi)**
- **Contatto diretto con azienda esterna**
- **Uso di GitHub**

CONTRO

- **Necessità di conoscenze preliminari per utilizzo di alcuni strumenti**
- **Grande quantità di tempo necessaria per completare al meglio i vari step**
- **Utilizzo di versione Xubuntu non recente**

RINGRAZIAMENTI

**A TUTTI PER
L'ATTENZIONE**

**Al prof. Narizzano
e
a WeSii s.r.l.
per l'opportunità!**

Studente:

De Luca Jacopo