Secondo Incontro con Bluewind s.r.l

Ora Inizio: 15:45 24/12/2024
Ora Fine: 16:51 Versione: 1.0

Redattori	Malik Giafar Mohamed
Verifica	Marco Perazzolo Luca Parise
Approvazione	Ion Cainareanu
Uso	Esterno

nextsoftpadova@gmail.com

Indice

1 Introduzione 3
2 Presentazione UI 3
3 Supporto multilinguistico del plugin 3
4 Modalità di installazione del plugin 3
5 Formato di importazione dei requisiti e persistenza dei dati 3
6 Esclusione di file 3
7 Requisiti prestazionali 4
8 Presentazione schema funzionamento backend 4
9 Conclusioni 4

1 Introduzione

Durante il colloquio abbiamo discusso con il proponente dell'azienda Bluewind s.r.l l'avanzamento dei progressi nella creazione del PoC in modo da avere un feedback sul lavoro svolto.
Sono anche stati chiariti dei dubbi su alcuni aspetti dello sviluppo del plugin.

2 Presentazione UI

Abbiamo presentato la demo dell'interfaccia grafica del plugin che verrà utilizzata per la consegna. Il proponente ha gradito la soluzione grafica proposta e ha suggerito alcune possibili modifiche per migliorarne l'usabilità, per esempio:

- la possibilità di mettere in pausa l'analisi dei requisiti;
- la possibilità di visualizzare l'andamento dell'analisi, visualizzando i requisiti mancanti e quelli già analizzati;
- la possibilità di contrassegnare i suggerimenti dell'analisi, come completati o da ignorare.

È stato evidenziato come durante il filtro dei requisiti nell'interfaccia utente sia più performante il filtraggio solo per nome file anziché per contenuto.

3 Supporto multilinguistico del plugin

Durante il colloquio è stato spiegato come per ora il plugin dovrà essere soltanto in lingua inglese.

4 Modalità di installazione del plugin

Nello sviluppo del plugin sono sorti dei dubbi riguardo la modalità di installazione da parte degli utenti finali. Il proponente durante il colloquio ha spiegato come non sarà necessario pubblicare il plugin sul marketplace delle estensioni di Visual Studio Code, ma sarà sufficiente fornire la repository con il codice sorgente ed eventualmente delle release su GitHub.

5 Formato di importazione dei requisiti e persistenza dei dati

Il proponente ha spiegato che i requisiti da analizzare che saranno forniti al plugin provengono da un file CSV esportato da un'altro sistema. Dopo aver mostrato degli esempi di file di requisiti e il progetto a cui fanno riferimento, il proponente ha suggerito delle migliorie al formato di importazione da noi definito.

È stato anche chiarito che avendo a disposizione solamente un file CSV con i requisiti, non è necessario implementare un sistema di persistenza dei dati per memorizzare i requisiti analizzati, ma basta salvare una copia del file di origine con i risultati delle analisi.

6 Esclusione di file

Durante lo sviluppo del PoC abbiamo riscontrato la necessità di escludere alcuni file dalla lista da analizzare per evitare che il plugin prenda in considerazione file non necessari per l'analisi, per esempio intere cartelle di librerie importate. È stato concordato con il proponente di adottare una strategia simile a quella di GitHub, ovvero mettere a disposizione un file con una lista di percorsi da ignorare. Il formato del file verrà definito in seguito.

7 Requisiti prestazionali

Discutendo di questo ambito con il proponente sono emerse le seguenti necessità:

- Il plugin deve essere in grado di gestire file di grandi dimensioni senza bloccare l'interfaccia grafica, permettendo all'utente di lavorare in background. Per arrivare a questo obiettivo, può essere necessario spezzare la parte di import e analisi dei requisiti in due parti, in modo da poter anche permettere all'utente di mettere in pausa l'analisi e riprenderla in un secondo momento.
  - in caso di file di requisiti molto grandi, un requisito desiderabile è la suddivisione automatica del file in parti più piccole, lavoro che può anche essere fatto manualmente dall'utente per il momento.
- Sono necessari anche dei meccanismi che avvertano quando il plugin sta sottoperformando, ad esempio quando l'analisi richiede troppo tempo a causa della connessione o delle prestazioni limitate di Ollama
- Il modello del plugin deve essere modulare e permettere di cambiare modello ad Ollama in modo semplice.

Il proponente ci saprà confermare in futuro se sarà possibile fornire delle infrastrutture per testare il plugin con modelli più pesanti.

8 Presentazione schema funzionamento backend

Abbiamo presentato uno schema di funzionamento del backend del plugin, il proponente ha apprezzato la chiarezza dello schema e ha suggerito alcune modifiche per renderlo più comprensibile. È stato suggerito l'utilizzo dei principi EARS come knowledge per l'analisi semantica dei requisiti. Per quanto riguarda invece l'ambito embedded, il proponente ha incoraggiato l'utilizzo dei manuali per i vari componenti hardware come knowledge per gli LLM, ma si ha un margine di tolleranza più ampio per la verticalità dell'ambito, dato che alcuni manuali possono essere molto voluminosi.

9 Conclusioni

L'incontro si è concluso positivamente, il proponente ha chiarito i nostri dubbi e si è dimostrato disponibile e coinvolto nello sviluppo di un plugin che possa soddisfare le proprie esigenze.