ByteOps.swe@gmail.com

Norme di progetto

Informazioni documento

Data	30/10/2023
Redattori	A. Barutta R. Smanio N. Preto
Verificatori	E. Hysa L. Skenderi D. Diotto
Amministratore	F. Pozza
Destinatari	T. Vardanega R. Cardin

Versione	Data	Autore	Verificatore	Dettaglio
1.0.0	31/10/2023	Nome autore	Nome verificatore	descrizione

Indice

ByteOps

November 2, 2023

Contents

1 Processi di supporto

1.1 Documentazione

1.1.1 Introduzione

La documentazione costituisce l'insieme di informazioni scritte che accompagnano un prodotto software, fornendo dettagli utili agli sviluppatori, distributori e utenti. Il suo obiettivo principale è agevolare il team durante il processo di sviluppo, tracciando e documentando tutte le fasi e attività che fanno parto del ciclo di vita del prodotto per migliorare la manutenzione e la qualità del risultato finale. Un elemento chiave è garantire chiarezza e concisione nelle regole redazionali, insieme a una struttura uniforme per tutti i documenti che verranno redatti, in modo da promuovere la coerenza e la familiarità ai documenti per chi li utilizzerà.

1.1.2 Primi approcci alla redazione di documenti e problematiche riscontrate

Per la composizione iniziale dei documenti richiesti per la candidatura, è stata sperimentato un'approccio che impiegava gli strumenti di Google Drive. Tale metodologia consentiva ai redattori di redigere agevolmente i documenti senza la necessità di padroneggiare la sintassi LaTeX, con l'intenzione di trasporre successivamente il contenuto in LaTeX una volta che fosse stato validato dai verificatori. Tuttavia, questo approccio ha suscitato problematiche, tra cui:

- Rischio di incoerenza tra il contenuto presente negli strumenti di Google Drive.
- Prolungato impiego di tempo per la riscrittura in LaTeX, dovuto alla necessità di un passaggio aggiuntivo.

Per tali ragioni si è presa la decisione di adottare un nuovo approccio.

1.1.3 Documentation as Code

L'approccio che si intende adottare è quello di "Documentation as Code" (Documentazione come Codice) che consiste nel trattare la documentazione di un progetto software allo stesso modo in cui si tratta il codice sorgente. Questo approccio è incentrato sull'utilizzo di pratiche e strumenti tipici dello sviluppo software per creare, gestire e distribuire la documentazione. Alcuni aspetti chiave di "Documentation as Code" includono:

- **Versionamento:** La documentazione viene gestita attraverso un sistema di controllo versione (come Git), consentendo di tenere traccia delle modifiche nel tempo e di ripristinare o confrontare versioni precedenti.

- **Scrittura in Formato Testuale:** La documentazione viene scritta utilizzando formati testuali leggibili da macchine (come Markdown o reStructuredText), che sono facilmente gestibili attraverso un sistema di controllo versione.
- **Automazione:** L'automazione è utilizzata per generare la documentazione in diversi formati (ad esempio, HTML, PDF) a partire dai sorgenti testuali. Gli strumenti come Sphinx, Jekyll o MkDocs sono comunemente utilizzati per questo scopo.
- **Collaborazione:** La documentazione è soggetta a revisione del codice e coinvolgimento collaborativo, proprio come il codice sorgente. Questo promuove la qualità e l'accuratezza della documentazione.
- **Integrazione Continua:** L'integrazione continua può essere utilizzata per automatizzare il processo di generazione e distribuzione della documentazione ogni volta che ci sono modifiche nel repository.
- **Distribuzione:** La documentazione può essere distribuita insieme al software, consentendo agli utenti di accedere facilmente alle informazioni aggiornate.

Questo approccio porta diversi vantaggi, tra cui una maggiore coerenza, una migliore tracciabilità delle modifiche e una facilità di manutenzione. Inoltre, il concetto di "Documentation as Code" si allinea con la filosofia DevOps, dove la collaborazione e l'automazione sono valori chiave.

1.1.4 Redazione documenti

Procedimento in sintesi
Per rispettare i principi di "Documentation as Code" l'approccio attualmente adottato richiede la competenza e l'utilizzo di LaTeX per la stesura dei documenti e una repository ad accesso limitato Git chiamata "Sorgente documenti" dove versionarli. I redattori, seguendo la struttura dettagliata nei prossimi paragrafi, redigeranno il contenuto dei documenti utilizzando il linguaggio LaTeX su un ramo dedicato nella repository "Sorgente documenti".
Al termine del loro lavoro, segnaleranno il completamento posizionando l'Issue relativa al compito di redazione del documento nella colonna "Documentazione da validare" della DashBoard documentazione di GitHub relativa alla repository pubblica "Documents" e generando una pull request . Questa colonna conterrà i compiti relativi ai documenti completati dai redattori, che necessitano ora della validazione da parte dei verificatori.
I file sorgenti LaTeX saranno quindi gestiti e versionati nella repository con accesso limitato denominata "Sorgente documenti", strutturata in modo identico alla repository pubblica chiamata "Documents" ma con file LaTeX al posto dei PDF.
I verificatori quindi troveranno il documento da validare in questa repository.

Quando i verificatori avranno convalidato il contenuto, dovranno seguire le istruzioni al paragrafo 1.1.5.2 per formalizzare la verifica.

1.1.5 Ruoli: procedure nel dettaglio

1.1.5.1 I redattori: compiti e casi di sviluppo

Il redattore incaricato della stesura di un documento è tenuto ad adottare un approccio simile a quello richiesto per apportare modifiche al codice sorgente di un software utilizzando il modello di branching di Git noto come "feature branch".
Di conseguenza, dalla repository "Sorgente documenti" contenente i sorgenti LaTeX, dovrà creare un nuovo branch Git in locale e passare ad esso mediante l'utilizzo del comando:

```
git checkout -b branch_identificativoDocumento
```

A questo punto, dopo aver creato il documento assegnato e avviato la stesura, affinché gli altri redattori possano continuare il lavoro, è necessario rendere il branch accessibile anche nella repository remota, seguendo i seguenti passaggi:

1. Assicurati di essere nel branch corretto:

   ```
   git checkout branch_identificativoDocumento
   ```

2. Aggiungi e committa le tue modifiche:

   ```
   git add .
   git commit -m "Descrizione del commit"
   ```

3. Pusha il tuo branch:

   ```
   git push origin branch_identificativoDocumento
   ```

La denominazione del branch utilizzato per la stesura del documento deve consentire un'identificazione rapida e chiara del documento stesso. Pertanto, adotteremo la seguente convenzione nella nomenclatura dei branch per la redazione dei documenti.

Convezione nomeclatura branch documentazione

- **Verbali interni**: `branch_VerbaleIn_dd_mm_yyyy`
- **Verbali esterni**: `branch_VerbaleEx_dd_mm_yyyy`
- **Norme di progetto**: `branch_NormeDiProgetto`

Caso modifica documento in fase di redazione
Qualora il redattore desideri proseguire la modifica di un documento precedentemente avviato o intendesse continuare la stesura di un documento iniziato da un altro redattore, è necessario:

1. Aggiornare la propria directory locale con i nuovi branch attivi:
   ```
   git pull
   ```
2. Spostarsi nel branch di redazione del documento
   ```
   git checkout branch_identificativoDocumento
   ```

Al termine del loro lavoro,i redattori segnaleranno il completamento della attiva a loro assegnata posizionando l'issue relativa nella colonna "Documentazione da validare" della DashBoard documentazione.
Inoltre dovranno attuare una **pull request**.
Si tratta di una richiesta di unire le modifiche fatte nel branch di redazione a un altro branch, in questo caso il main.

1. Vai sul repository su GitHub
2. Apri la sezione "pull request"
3. Clicca "New pull request"
4. Selziona come branch di destinazione "main" e come branch sorgente il ramo utilizzato per la redazione del documento da validare
5. Clicca "Create pull request"
6. Dai un titolo e una breve descrizione alla pull request
7. Seleziona i verificatori su "Reviewers"
8. Clicca "Create pull request"

Se i verificatori non convalidano il documento, riceveranno feedback allegati alla pull request relativi ai problemi identificati.

1.1.5.2 I verificatori: compiti e casi di sviluppo

Il ruolo del verificatore nei documenti è cruciale per garantire la qualità e l'accuratezza del contenuto.
Quando il verificatore individua un'Issue nella colonna "Documentazione da validare" della DashBoard documentazione, sarà tenuto a convalidare il file corrispondente presente nella repository con accesso limitato denominata "Sorgente documenti".
In aggiunta, il revisore riceverà una notifica via email quando il redattore completa la propria attività, comunicandogli la presenza della pull request assegnatagli.
Nella sezione "pull request" di GitHub, il revisore troverà la richiesta di unire il branch di redazione al branch "main", assumendo il ruolo di revisore. Accedendo alla pull request su GitHub, il revisore avrà la possibilità di esaminare attentamente il documento in questione e di aggiungere commenti visibili ai redattori nel caso in cui siano necessarie modifiche per la validazione.

Per validare un documento i le verifiche da attuare sono:

- **Revisione della Correttezza Tecnica:** Revisione tecnica del documento per garantire che tutte le informazioni siano corrette, coerenti e rispettino le norme stabilite.
- **Conformità alle Norme:** Verifica che il documento segua le linee guida e gli standard prestabiliti per la formattazione, la struttura e lo stile.
- **Consistenza e Coerenza:** Si assicura che il documento sia consistente internamente e coerente con altri documenti correlati. Verifica che non ci siano discrepanze o contraddizioni.
- **Chiarezza e Comprensibilità:** Valuta la chiarezza del testo, assicurandosi che il linguaggio sia comprensibile per il pubblico di destinazione e che non ci siano ambiguità.
- **Revisione Grammaticale e Ortografica** Controlla la grammatica, l'ortografia e la punteggiatura del documento per garantire una presentazione professionale.

Dopo l'attuazione dei controlli sopra citati e verificato il loro rispetto, i passi per convalidare il documento sono i seguenti:

1. **Apri la Pull Request:**Accedi alla pagina della pull request in cui agisci come revisore nel repository "Sorgente documenti" su GitHub
2. **Accettare la pull request:** Cliccare il pulsante "Merge Pull Request".
3. **Sposta la issue in Done:** Nella DashBoard documentazione sposta la issue relativa al documento validato da "Documentazione da validare" a "Done".
4. **Genera il PDF:** Compila il file Latex e genera il PDF

5. **Carica PDF:** Inserisci il PDF generato dal passaggio precedente nella repository Documents nella corretta posizione.
6. **Elimina il branch di redazione:** Nella repository "Sorgente documenti" apri la sezione "branches" ed elimina il branch relativo alla redazione del documento appena validato.

1.1.6 Verbali: struttura generale

Questo documento assume l'importante ruolo di costituire un registro ufficiale, atto a riportare in maniera accurata gli argomenti trattati, le decisioni adottate, le azioni da intraprendere e le figure coinvolte.
In particolare, è possibile distinguere tra verbali interni, destinati all'uso all'interno di un gruppo o di un'organizzazione, e verbali esterni, che trovano applicazione quando vi sono terze parti coinvolte nelle discussioni o nelle decisioni documentate.
Nella prima pagina vengono specificate:

- Nome e mail del gruppo
- Data del documento
- Tipologia della riunione
- Luogo: canale di comunicazione adottato
- Ora di inizio e fine dell'incontro
- Ruoli
- Destinatari
- Partecipanti della riunione

In questo tipo di documento non è presente il versionamento del file in quanto non necessario.

1.1.6.1 Verbali interni

Un verbale interno è un documento redatto all'interno del nostro gruppo con lo scopo di registrare e documentare in maniera dettagliata i particolari di una riunione che coinvolge esclusivamente i membri del nostro team.
La sua funzione primaria consiste nel registrare in modo preciso le conversazioni tenutesi durante l'incontro, le decisioni prese e i piani futuri concordati.
L'audience principale di un verbale interno è costituita dai membri dell'organizzazione o del team.

1.1.6.2 Verbali esterni

Un verbale esterno è un documento scritto utilizzato per registrare e documentare in modo accurato i dettagli di una riunione o un incontro che coinvolge non solo i membri interni al nostro gruppo, ma anche terze parti esterne ad esso.
Il pubblico principale di un verbale esterno può includere individui o entità al di fuori del team, come clienti o l'azienda proponente.
La principale distinzione rispetto ai verbali interni risiede nel fatto che entrambe le parti coinvolte nell'incontro convalidino il verbale e concordino sul suo contenuto. Pertanto, è essenziale stabilire un metodo di validazione del documento che garantisca la correttezza e l'accuratezza delle informazioni registrate.
Nel nostro caso, [specificare qui il metodo di validazione, ad esempio, la firma delle parti coinvolte, l'approvazione via e-mail, o qualsiasi altra procedura stabilita contrattualmente] serve a confermare e legittimare il contenuto del verbale, contribuendo a evitare fraintendimenti e dispute future.

1.2 Canali di comunicazione

1.2.1 Meeting settimali

Chi fa cosa? quando? problemi riscontrati, cosa mi impedisce. Tipo stand up meeting