28/02/2022: Introduzione ed Issue Tracking System

Il corso tratta principalmente la condivisione e creazione di software gestito in maniera condivisa. In questo senso, il programma deve essere controllato nel suo codice sorgente e poi integrato attraverso la fase di build. Lo sviluppo avviene attraverso una modalità di integrazione continua, al fine di verificare se sia effettivamente funzionante in vari ambienti di sviluppo.

In linea di massima si segue il principio della *continuous delivery pipeline*, che è un'implementazione del paradigma continuo, in cui build, test e distribuzioni automatizzati sono orchestrati come un flusso di lavoro di rilascio. Tutto ciò fa parte della creazione Agile dei progetti, al fine di gestire vari processi tradizionali e fornire molteplici e nuove funzionalità. Tutto ciò viene garantito attraverso vari principi, prendendo ad esempio l’esplorazione continua dei bisogna del mercato/clienti attraverso feedback, integrazione continua di caratteristiche, creazione e consegna di fasi di produzione e rilascio a richiesta, rendendo il prodotto disponibile una volta correttamente pronto.

Gli esempi di build tracking sono il bugtracker di Ubuntu, con diverse migliaia di bug giornalmente segnalate. Segue Jira, software molto utilizzato, dove si riportano le segnalazioni, menù e a destra la versione del software in cui un certo bug è stato sistemato. Anche GitHub presenta una serie di tag ed è un buon issue tracking system (trovare quindi le criticità, mettendo nero su bianco quante più cose possibili, utile per tutti gli attori di un progetto software). Questi sistemi mantengono una lista di problemi (issues), permettendo di trovare o localizzare una serie di problemi e permettendo di risolvere anche a catena i problemi (per esempio tramite servizio clienti). Un esempio di questo sono i bugtrackers, vedendo se c’è già un bug di quel tipo.

Abbiamo per esempio *l’ITS per la gestione di Progetto*, che facilita il management, lo sviluppo e il ciclo di vita di un certo progetto, per esempio anche in base alle richieste del cliente, segnate come requisiti. Quando nasce una certa attività, dopo una fase di analisi, si usa una stima per capire quanto tempi si impiega per certe attività. Si devono usare il più possibile, perché magari lo sviluppatore non lo fa e lo stesso project manager dovrebbe impiegarci di più; soprattutto in un contesto reale risulta quììuna perdita di tempo effettiva.

Tutto questo serve nelle riunioni col cliente, tra team di sviluppo e Project Manager, cosiddetta SAL (Stato Avanzamento Lavori). Sicuramente tutto questo è una misura della qualità, capendo dove effettivamente si è verificata una certa problematica/bug e avendo quindi istantanee. È importante dare una giusta priorità alle singole attività, quantificando il tempo impiegato singolarmente e capendo l’effettiva efficacia. I Work Item (quindi oggetti di lavoro in un progetto) devono avere una chiara assegnazione, riportando assegnatario e segnalante.

Ottimi prodotti in questo senso Jira, Github, ecc. *I Work Item* sono degli strumenti utili per capire come sviluppare e lavorare su un progetto, magari dal punto di vista del cliente facendo capire le sue esigenze e come usarlo. Una cosa utile in questi strumenti è capire la situazione di un bug, riproducendolo e capendo i passi che hanno portato alla sua scoperta/segnalazione/risoluzione. Sempre in merito ai work item, esistono vari campi identificativi, come il progetto di riferimento, un codice univoco, un tipo (campi, stati, schermate, workflow), stato nel workflow, priorità, stato di risoluzione, versione di riferimento dove è stato trovato il bug.

Vi sono poi i campi come le etichette di classificazione, collegamenti tra i vari Work Item, assegnatario/responsabile, segnalante, date di ultimo aggiornamento/risoluzione/stima delle attività, tempo speso ed eventuali allegati.

Per portare a compimento le attività, l’insieme di stati e transizioni dei Work Item e del loro tempo di vita è il *Workflow*, insieme di stati e transizioni di un Work item e che, associato ad un progetto, ne ricostruisce lo storico delle segnalazioni, tracciandone soluzione/risoluzione. Determinate esigenze del cliente sono documentate negli *SLA (Service Level Agreement)*, per esempio la risoluzione di una certa problematica X entro un certo tempo Y. Ciascun requisito può essere categorizzato e classificato in *macrorequisito*, *requisito* e *sottorequisito*, suddividendo e specificando l’individuazione e l’utilizzo delle sottoattività.

Importanti in questo senso sono i *collegamenti*, definendo le relazioni tra i Work item, solitamente bidirezionali e utilizzate come possibile criterio di ricerca, capendo o meno le relazioni tra i componenti.

Altro strumento utile sono le *notifiche*, che segnalano la risoluzione e fungono da tracciamento. Citiamo anche i *filtri*, i quali possono essere salvati ed esportati; in generale sono quindi una base per creare report, board e dashboard. I report sono suddivise in tipologie e caratterizzate graficamente da vari tipi di grafico (torta, istogramma, etc.). Utili anche le *board/bacheche*, visualizzando i work item di uno o più progetti e visualizzandone l’insieme e le interazioni tramite possibili filtri e interagendo velocemente con ognuno di essi.

Un buon ITS deve essere configurato identificando i *processi*, quindi possibili best practices definiti dai framework usati oppure regolati dallo stesso cliente e identificare/configurare gli strumenti utili, definendo tipi, campi custom, work item e collegamenti, tracciando quindi le informazioni in tempo breve.

Esso normalmente inizia, viene pianificata una versione, vengono risolte attività (fase di sprint, nell’esempio del caso d’uso Jira) e poi rilasciata una versione. Nel team, in base all’ITS il *manager* (amministratore) definisce, una volta creato il progetto, il processo da seguire, con relativi work item, eseguendo e censendo un modello di stima e aggiungendo gli specifici utenti, assegnando loro i giusti permessi. Nel team di sviluppo gli utenti (intesi come team di sviluppo) selezionano i work item, avviano e completano la lavorazione documentando le attività e programmando il rilascio.

Il manager monitora l’avanzamento, definendo le nuove versioni ed iterazioni, dando report specifici al cliente.

Tra i benefici di utilizzo di un ITS si ha l’implementazione di processi, verificandone l’adozione, misura di qualità e soddisfazione del cliente, produttività del team riducendo sprechi e spese.