



Carbon12 – Predire in Grafana

Verbale Esterno

Informazioni sul documento

Versione	1.0.0
Data di creazione	2020/02/14
Redazione	Francesco Gobbo
Verifica	Nicolò Fassina
Uso	Interno
Destinatari	Carbon12

1 Informazioni sull'incontro

Data	2020/02/14
Luogo	Dipartimento di Matematica "Tullio Levi-Civita" - Università degli Studi di Padova
Ora inizio	16.30
Ora fine	17.15
Partecipanti	Manuel De Franceschi Nicolò Fassina Francesco Gobbo Andrea Longo Alessandro Lovo
Partecipanti esterni	Tullio Vardanega

2 Ordine del giorno

1. Metrica di versionamento documenti e prodotto
2. Capitolo dell'appendice B (Attuazione dei rischi) del PdP
3. Discussione sul modello incrementale
4. Cruscotto attività di verifica del PdQ
5. Progettazione da sviluppare per il PoC
6. PoC

3 Discussione Argomenti

3.1 Metriche di versionamento documenti e prototipo

L'incontro ha avuto inizio con l'analisi dell'attività di versionamento, perché ritenuta fondamentale per il gruppo in quanto rappresenta l'applicazione del metodo incrementale nei documenti.

Il professore ha suggerito come definire l'indice partendo da degli esempi: Netflix, Gmail e Chrome. Per quanto riguarda Netflix, ha spiegato che c'è un costo certo e un ritorno o incasso in funzione della bontà del software. L'obiettivo è avere un software buono, che è in uso se è buono, e per raggiungere questo obiettivo fanno uso di una solo repo e tramite una build automatica il contenuto viene aggiornato con la nuova versione. Tale versione è sempre buona, perciò all'interno della repo c'è solo software buono.

Ha poi spiegato che il principio della repo buona viene applicata anche per Gmail (versione browser): in quanto all'accesso viene istantaneamente usata l'ultima versione rilasciata, perciò il contenuto della repo deve essere solo e unicamente prodotto buono, perciò solo prodotto verificato. Nella repo si entra solo se si è "buoni", dunque il suo contenuto è composto di prodotti che hanno superato tutte le attività di verifica.

Per quanto riguarda lo scatto del numero di versione, esso non avviene solo per aver fatto un'attività: il fare non è dunque un avanzamento di versione. Lo scatto di versione del prodotto è sconnesso dalla build, infatti questa può avere un valore totalmente a sé stante rispetto al precedente rilascio.

Inoltre, lo scatto di versione non ha tutto allo stesso livello, per esempio l'indice di versione di Chrome è un numero composto da una combinazione di numeri: un numero major, un medium e un minor, affiancati da un numero di build. Build si incrementa ogni volta che avviene una build (non tutte sono necessariamente visibili), mentre gli altri numeri indicano cosa è cambiato rispetto alla versione precedente.

L'ultimo aspetto da considerare è il rapporto tra documenti e progetto: in quanto i documenti sono una parte del progetto, perciò bisogna stabilire l'associazione dei documenti rispetto al risultato finale, non bisogna avere una visione segregata, ma una visione complessiva dell'avanzamento del progetto, in quanto tutto farà parte del prodotto.

I membri del team, dopo l'incontro, hanno ipotizzato la seguente indicizzazione da applicare ai documenti e al prodotto:

[Indice stato avanzamento Prodotto].[Indice rilascio revisione].[Indice versione approvata]-[Build]

- Indice stato avanzamento Prodotto = $\{0, \alpha, \beta, 1\}$

Questo numero indicherà a che stato di avanzamento siamo arrivati rispetto al raggiungimento del prodotto finale:

- 0: indica che siamo nelle fasi iniziali senza aver ancora delineato bene le effettive caratteristiche del prodotto. Questo valore ci accompagnerà in linea generale fino alla Revisione di Progettazione.
 - α : indica invece che stiamo iniziando lo sviluppo del prodotto, ma che ciò che si è realizzato fino a quel momento non ha ancora raggiunto i requisiti sufficienti per essere definito un prototipo del prodotto desiderato.
 - β : indica invece che ciò che si è prodotto si avvicina al prodotto desiderato, rendendolo un simil prototipo.
 - 1: indica che il prodotto è completo e pronto per essere usato.
- Indice di rilascio revisione = $\{0,1,2,3,4\}$
Questo numero ci permette di indicare a quale rilascio si è arrivati, in quanto ci indica la baseline di avanzamento rispetto agli obiettivi prefissati nel PdP. Per essere più precisi tale numero ci indica l'accesso alle revisioni di avanzamento fissate dal cliente, in quanto se ciò che si è realizzato fino ad ora soddisfa il cliente, tale risultato permette al team di definire un proprio avanzamento per il raggiungimento dell'obiettivo finale (il completo soddisfacimento del cliente).
 - Indice versione approvata
Questo numero indica che la versione del documento o software realizzato è stato definitivamente validato e approvato, perciò i suoi contenuti sono buoni.
 - Build
Questo numero, che il team deciderà se inserirlo o meno nella tabella delle versioni per i documenti, indica gli incrementi e le attività di verifica effettuate dal team.

3.2 Capitolo dell'appendice B (Attuazione dei rischi) del PdP

Il prof ha spiegato che il problema nell'appendice "Attuazione dei rischi" sta nel fatto che il PdP serve principalmente al team, in quanto è concettualmente un diario che funziona come una pila, cioè in cima c'è la situazione attuale dei costi.

Il PdP serve dunque a difendere il nostro lavoro, serve per indicare cosa ci manca e cosa si è sviluppato. Perciò non bisogna aggiornarlo solo per le revisioni, bisogna aggiornarlo quotidianamente (idealmente, perciò è sufficiente aggiornarlo spesso). Bisogna vederlo come strumento interno per vedere i problemi che emergono e rinnovare la lista dei problemi quando vengono risolti.

I contenuti devono perciò essere aggiornati, così che, quando il documento viene consegnato all'esterno ci siano solo contenuti veri.

3.3 Discussione sul modello incrementale

Procedere per incrementi è una buona soluzione, perché non fallisce (cosa che succede con l'Agile a discapito del cliente), perché il team ci arriva imparando, allocando risorse, con incrementi piccoli, ad una frequenza ragionevole. Queste sono le basi del modello.

Gli incrementi servono al proponente, non al cliente. Perciò i piccoli incrementi servono al team per vedere come procedere e come si sta procedendo, ma anche al proponente per consigliare una realizzazione che possa poi soddisfare il cliente, perché a quest'ultimo interessa solo ciò che si consegna alla revisione.

Prendendo come caso il PoC, esso lo si sviluppa per il proponente, in quanto il team si mette in gioco per vedere la correlazione tra i vari strumenti. È sconsigliato creare un unico PoC, poiché è rischioso fare tanto lavoro senza ricevere alcun tipo di approvazione su come si sta procedendo, diventa perciò più conveniente creare vari "PoCini" che permettano al gruppo di capire l'interazione degli strumenti e se c'è intesa con il proponente, questi verranno combinati per realizzare il PoC.

Operando così per piccoli incrementi, la pianificazione con il proponente è d'obbligo, poiché ci permette di capire gli incrementi necessari per soddisfare gli obiettivi in breve periodo e i colloqui con il proponente permettono al team di risolvere i problemi che emergono.

Il prof ci ha consigliato di imporci una maggior consapevolezza di quanti incrementi servano, per soddisfare uno sviluppo crescente. Gli incrementi sono da stabilire tra i membri del team e il proponente, poiché nel dialogo si capisce cosa è interessante da sviluppare e se il ritmo per lo sviluppo è sostenibile dal team. Per il proponente è interessante come ci avviciniamo nello sviluppo degli incrementi, perciò un dialogo costante tra team e proponente giova ad entrambi per il raggiungimento di un buon risultato.

3.4 Cruscotto attività di verifica del PdQ

Il prof ha spiegato cosa intendesse con cruscotto partendo dall'esempio calcistico della classifica della serie A: l'obiettivo non è vedere come sono posizionate le squadre, ma vedere come sta procedendo una squadra. Cioè bisogna osservare lo sviluppo storico: se una squadra, sebbene in cima alla classifica, continua a rimanere su una costante, ha sicuramente meno rilevanza rispetto ad una squadra che sta vincendo partite su partite, risalendo così la classifica.

Quindi il cruscotto che dà le tendenze è molto più rilevante rispetto ad una serie di dati statica che conferma uno stato buono. Ciò che è interessante da rilevare sono i requisiti soddisfatti, il coverage rispetto al branch, o i ticket risolti, quindi ciò che è fotografabile ad ogni istante e mostra un senso di controllo nello sviluppo e nell'attività lavorativa.

3.5 Progettazione da sviluppare per il PoC

Si è poi chiesta una delucidazione in merito al livello di progettazione da presentare alla consegna del PoC.

Il PoC da consegnare per l'RP serve per farci "toccare" le tecnologie con cui lavoreremo e come collegarle con un minimo senso, diventa perciò un tentativo di metterle insieme. Tuttavia, se puntiamo ad ottenere un PoC incrementale, non può non possedere un minimo di progettazione alla base.

Quindi non dobbiamo definire il “PoCino” come un risultato, perché il cliente non ha grosse aspettative sull’implementazione di qualcosa che valga per il futuro, però il team ha stabilito come modello quello incrementale, perciò una progettazione minima alla base dovrebbe esserci.

3.6 PoC

Il software non è solo codifica, ci deve essere alla base un po’ di consapevolezza della struttura, ovviamente non ci si aspetta per il PoC una progettazione definitiva, ma è sufficiente un quadro di progettazione per capire cosa fa il software.