

3° Evento Metrico 2017 (Caserta, 01/12/2017) – Abstract Presentazioni

Misura dei servizi condivisi tra applicazioni ([F.Di Cola](#) - SOGEL) [presentazione valida per il programma IFPUG CEP – Certification Extension Program]

I Servizi e la loro definizione non per riutilizzare alcune logiche funzionali o per soddisfare requisiti non funzionali sono il pane quotidiano in molte aziende, soprattutto in quelle dove il parco software è di una certa dimensione. La presentazione, partendo da un parallelo tra la definizione di servizio SOA e le definizioni del CPM riguardo processi elementari e dati condivisi, illustra una proposta di misura dei singoli servizi, ove essi si possano considerare derivati dai requisiti funzionali. Si cerca inoltre di contestualizzare le casistiche in cui i servizi possono effettivamente essere misurati come “funzionali” e quindi secondo le regole del CPM.

IFPUG in Agile ([T. Fehlmann](#) – E-P-O)

Una banca in Germania non aveva esperienza con i Function Points ma voleva conoscere la propria produttività. Utilizzando i modelli “Transaction Map” che ho presentato l’anno scorso, il conteggio era molto più facile, rapido e efficace tanto quanto con metodi tradizionali, ma anche come la variante Gartner FFPA. Gli estratti del modello per ogni Sprint si fa quasi in modo automatico. L’unica documentazione affidabile era data dalle User Stories, e qualche documento architeturale. Il metodo più affidabile era quello di creare un modello della vision/strategia, dunque dei concetti iniziali del prodotto software, e poi sprint per sprint con le user stories. Le user stories avevano sempre attore, punto di vista, riferimento ai dati di interesse, e qualche attributo qualitativo, con un buon livello qualitativo. La comparazione dei FP per Sprint con gli Story Point per Sprint, e con le ore rapportate ha dato indicazioni di produttività utili che andiamo presentare.

Perché adottare i modelli di qualità dei prodotti ISO 25000 ([D. Natale](#) – UNINFO)

Alla luce di recenti esperienze internazionali la presentazione mira a descrivere il modello di qualità ISO del software, dei dati e dei servizi IT, sviluppati nell’ambito del progetto ISO SQuARE. Saranno descritte alcune esperienze e “lessons learned” relative all’applicazione dei modelli in Italia e in Europa. Infine sarà dato un quadro sintetico inerente la Guida nazionale sull’applicazione della misurazione della qualità dei dati, che si sta mettendo a punto nell’ambito della Commissione UNINFO “Ingegneria del software”.

Il ‘Ballon effect’: ovverosia cosa mi hai portato a fare a Caserta se non misurassimo i progetti... ([L. Buglione](#) - Engineering)

Uno degli elementi componenti lo schema ABC è la sequenza ‘Q→T→C’, ovverosia il flusso logico da seguire per una stima che diventa poi un conteggio: determinare le quantità (Q) delle componenti (asset) da considerare, derivare i tempi di lavoro (T) in termini di effort e duration sulla base delle produttività (Q/T) storiche di un’organizzazione, derivare i costi/corrispettivi (C) finali in base ai valori di Q e T derivati. Cosa succede se però non si definiscono e misurano correttamente le quantità di interesse? Un FP è una misura di progetto o di prodotto? Uno Stress Test o un Test di Integrazione entra o meno nei calcoli per determinare quanti FP/giorno considerare per le stime di una change request/progetto oppure no? Perché spesso un’organizzazione – indipendentemente se committente o fornitore – sbaglia le proprie stime? La presentazione illustrerà alcuni dei principali paradossi numerici che rischiano di avvenire quotidianamente nei contratti ICT pubblici e privati, partendo da un’analisi EAM (Entità-Attributo-Misura) di un tipico progetto.

Esperienza sul governo metrico delle forniture Agile ([E.Scopece](#), [I.Mazzacano](#), [A.Bei](#), [P.Franco](#)– BNL/CapGemini)

Il Gruppo BNP Paribas, ha lanciato nel 2014 l’implementazione di un modello globale di sourcing per servizi di ADM (Application Development e Maintenance) con obiettivi di industrializzazione, aumento della qualità e riduzione dei costi. Nel contesto di tale iniziativa globale BNL ha stipulato con Capgemini un contratto locale. In particolare il servizio di Application Development è regolato contrattualmente attraverso tre elementi: (1) Un modello di sourcing scope che consente di specificare attività e deliverable in relazione alle esigenze del progetto; (2) Un processo di stima e consuntivazione del size del software basato su Gartner FFPA; (3) Una produttività che incrementa di anno in anno secondo una roadmap quinquennale predefinita. Nel 2014 BNL inizia ad adottare la metodologia Agile e nasce l’esigenza di adattare le rigorose regole contrattuali concepite per metodi Waterfall ad un approccio allo sviluppo iterativo flessibile ed adattivo. In questa presentazione si descrive il processo utilizzato per l’analisi dei gap contrattuali rispetto alla nuova metodologia, la definizione delle regole e dei parametri per colmare i gap, il piloting e la conseguente valorizzazione dei parametri. Si descrivono inoltre gli interventi sui 3 elementi sopra citati: 1. l’adattamento del Sourcing Scope: attuato mediante un mapping delle attività Waterfall con quelle tipiche dei cerimoniali Agile e la rilevazione degli impegni medi previsti per ciascuna attività; 2. l’adattamento del processo di stima e conteggio al modello e ai cerimoniali agile; 3. l’adattamento della produttività contrattuale non più fissa ma resa parametrica da una matrice di complessità che tiene conto delle peculiarità del progetto.

Understanding the Error Behaviour of Complex Critical Software Systems ([R. Della Corte](#) – Univ. Federi co II - Napoli) [Premio Studenti 2016]

Understanding the error behavior of software systems consist in obtaining knowledge on error types, intermediate paths and effects pertaining activation of faults. That knowledge allows gaining insights into the behavior of software systems under errors, which is of paramount importance to design/develop dependable software. For example, it drives the design and placement of error detection/recovery mechanisms, highlights the components most exposed to errors, allows understanding how components affect each other under errors and establishing where/what error types are likely to cause failures. The analysis of system-generated data, supported by the evaluation of ad-hoc metrics, represents a valuable mean to understand the error behavior of software systems. A variety of

techniques, such as source code instrumentation, variable tracing and white-box approaches, have been used so far to this aim; although valuable, they entail a degree of system internals knowledge, visibility and intervention that cannot be pursued for complex critical software systems. This presentation proposes a method and innovative metrics to characterize the error behavior of complex critical software systems. Differently from many existing approaches, the proposal capitalizes on the use of field data, i.e., naturally-emitted data by software systems in production environments, such as logs and assertions, aiming to support quantitative assessments of the error behavior of software systems and the effectiveness of field data at reproduce that behavior. The method is based on the construction of error propagation graphs, and leverages the proposed metrics: (1) Error Determination Degree, which evaluates the ability of field data to infer the cause and effect of occurred errors, i.e., the related fault and failure, respectively; (2) Error Propagation Reportability, which assesses their ability at reporting the propagation of errors; (3) Data Dissimilarity, which evaluates their utility for troubleshooting purposes. The proposal has been used with data from logging, assertion checking, and source code instrumentation of two real-world Air Traffic Control software systems. The use of the method and the analysis of the metrics revealed failure reporting ability of data sources, error models they covered, propagation paths peculiarities, feedback for placing error detection/recovery mechanisms, and allowed comparing data sources, highlighting their orthogonality at reporting failures and errors propagation.

Metrics for Effort/Cost Estimation of Mobile Apps Development (G. Catolino – Univ. Salerno) [Premio Studenti 2016]

Le risorse o l'effort necessario per sviluppare un progetto software sono da sempre complicati da stimare in fase iniziale. La software effort estimation è un'attività chiave del project management necessaria per la pianificazione di un progetto software, stima delle risorse e controllo della qualità. I metodi di stima dell'effort possono essere suddivisi in metodi model-based e non model-based. I metodi non model-based utilizzano l'esperienza umana per prevedere l'effort di un progetto, mentre i metodi model-based utilizzano diversi algoritmi al fine di prevedere l'effort di nuovi progetti. Tali modelli vengono generalmente costruiti utilizzando Function Point o COSMIC Function Point come predittori. Tuttavia la maggior parte delle metriche utilizzate per la predizione dell'effort, incluse quelle sopra indicate, possono essere utilizzate quando i requisiti di un sistema sono già stati specificati o il ciclo di vita del software è già iniziato. In questa tesi abbiamo affrontato questo problema, applicando una metodologia ispirata a quella definita da Mendes et al.—nel contesto della stima dell'effort di applicazioni web—per definire una serie di metriche di “early effort estimation” in grado di stimare l'effort durante una fase iniziale di sviluppo nel contesto delle applicazioni mobile. Una volta definite le metriche, abbiamo indagato quante di esse potevano essere rilevanti in una fase successiva di stima dell'effort. Per questo motivo abbiamo condotto uno studio di mappatura in cui abbiamo cercato di trovare la relazione tra le metriche definite in questo lavoro e un insieme di linee guida per il “quick sizing” dei COSMIC Function Point per applicazioni Android proposti da Cozzolino et al. Infine è stato condotto uno studio empirico con l'obiettivo di valutare se le metriche presentate possono essere utilizzate per approssimare la dimensione finale delle applicazioni mobile da sviluppare (utilizzando l'output della mappatura). I risultati hanno mostrato che in tutti i casi le stime fornite dalle nostre metriche sono risultate molto vicine ai valori reali. Ciò significa che è effettivamente possibile fare affidamento sulle nostre metriche per ottenere una stima della dimensione e dell'effort necessario per sviluppare un'applicazione mobile.