



PROPOSIZIONE E SPERIMENTAZIONE DI UN PROCESSO AGILE ORIENTATO ALLA QUALITÀ DEL CODICE

Anno Accademico 2015/2016

Relatore: Danilo Caivano

Laureando: Gianluigi Papadia

Dipartimento di Informatica - Università degli Studi di Bari

Via Orabona, 4 - 70125 - Bari

Tel: +39.080.5443270 | Fax: +39.080.5442536

serlab.di.uniba.it

Il contesto del lavoro di tesi

- ⇒ Il lavoro è stato svolto presso LinksMT S.p.A., azienda di consulenza specializzata nell'offerta di soluzioni e servizi IT nei settori PAL, Banking e Finance



Scenario

- ⇒ Storicamente, a partire dagli anni 60, la qualità di un prodotto software è stata perseguita puntando alla qualità dei processi di sviluppo e manutenzione utilizzati
- ⇒ L'assunto di fondo è che la qualità di un prodotto software:
 - ❑ È strettamente legata alla qualità del processo
 - ❑ Dovrebbe sempre essere definita e perseguita lungo tutte le fasi di produzione e manutenzione e non limitandosi esclusivamente alla valutazione del prodotto ultimato

Motivazioni (1)

La qualità del processo assicura la qualità del prodotto?

CORRIERE DELLA SERA

PROBLEMI TECNICI

Londra, spazio aereo chiuso (e poi riaperto) per guasto informatico

affaritaliani.it  *Il primo quotidiano digitale, dal 1996*

Disastro metrò Roma. Ecco i numeri e la verità sui guasti infiniti Atac

"...si è verificato un riempimento della memoria ed è probabile sia riferibile a un problema di software..."

sky TG24 HD

Fiumicino, mattinata di caos per un guasto informatico

IL TIRRENO EDIZIONE PISA

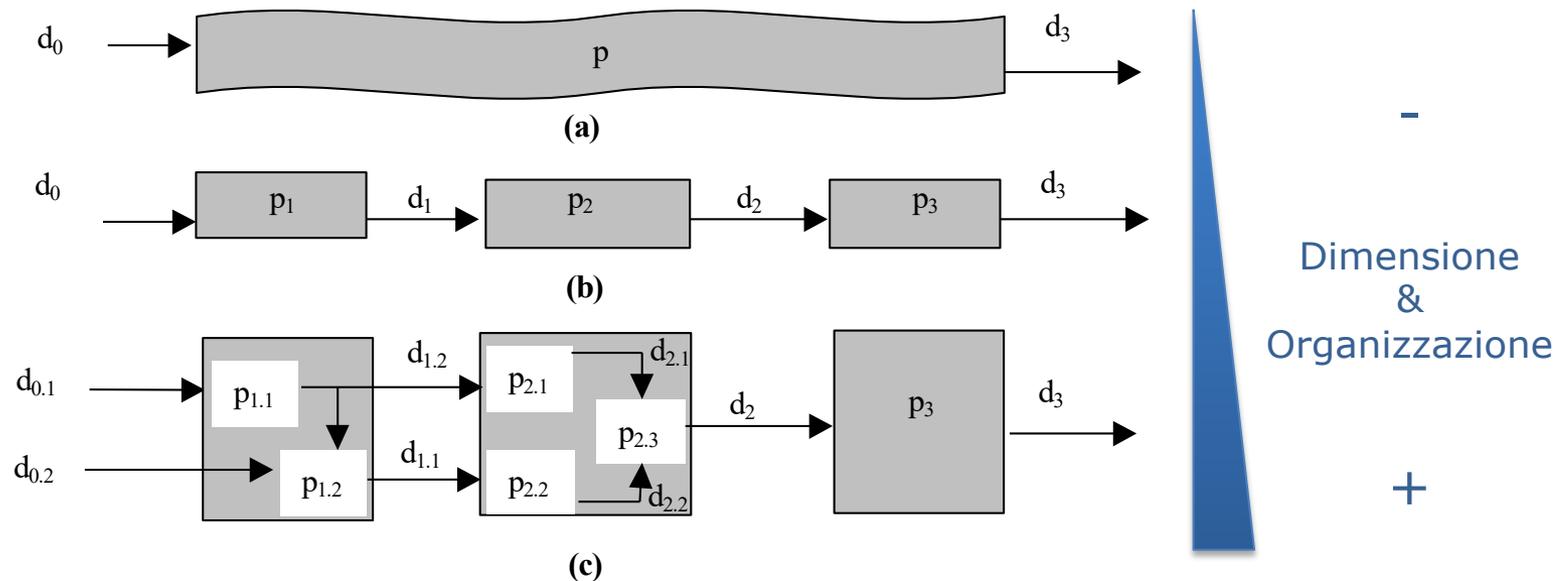
SANITÀ

Cisanello, risolto il guasto al sistema informatico

Disagi per le prenotazioni e le analisi non urgenti

Motivazioni (2)

- ⇒ I processi sono influenzati dalla dimensione, maturità, esperienza specifica su tecnologie ed ambienti.
- ⇒ In una PMI i processi in uso sono spesso “opachi” e non sempre sufficientemente formalizzati per consentire una valutazione quantitativa.



Le tendenze

- ⇒ Negli ultimi anni sono emersi una serie di approcci «leggeri» che focalizzano l'attenzione sul prodotto software:
- ❑ Euristiche, best practises e linee guida di programmazione sono ormai tali da poter esser verificate attraverso un'analisi statica del codice sviluppato



Scopo

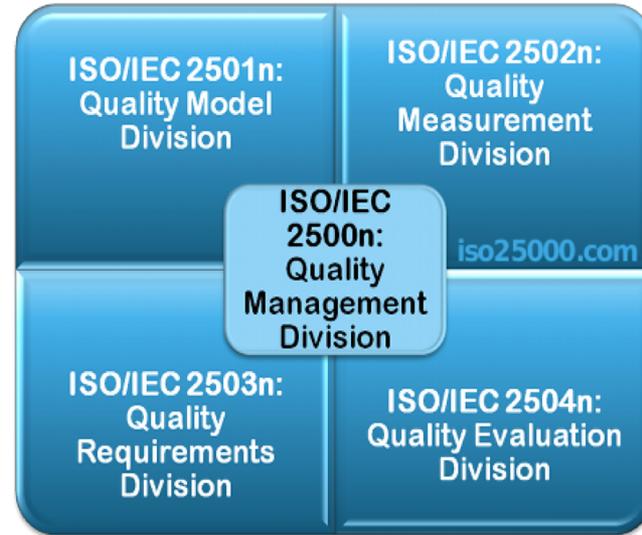
- ⇒ Proposizione di un approccio allo sviluppo software:
 - ❑ Basato sull'impiego di processi agili
 - ❑ Orientato alla qualità del codice prodotto
- ⇒ Strumentazione del processo proposto attraverso tools di supporto
- ⇒ Sperimentazione della proposta nell'ambito di progetti di interesse industriale

Processo proposto



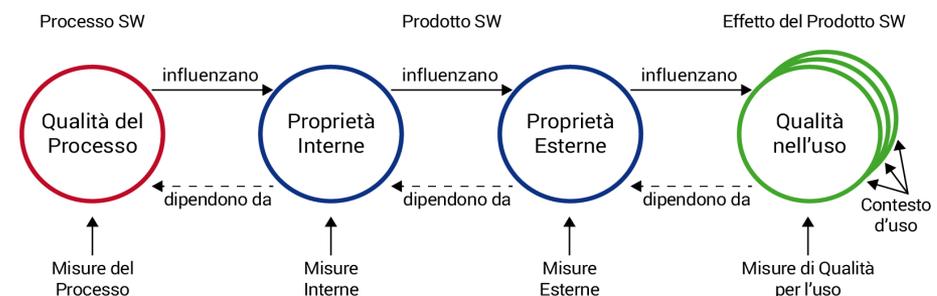
ISO/IEC 25000

⇒ Lo standard ISO/IEC 25000 ha l'obiettivo di creare un quadro di riferimento per la valutazione della qualità del prodotto software. Esso si compone da 5 elementi:



ISO/IEC 25010

- ⇒ Oggi la tendenza è quella di far convergere la valutazione verso un modello integrato basato su 3 punti di vista (ISO 25000):
- ❑ **QUALITA' IN USO**: esprime l'efficacia ed efficienza con cui il software serve le esigenze dell'utente, ed è correlata alla percezione diretta dell'utente
 - ❑ **QUALITA' INTERNA**: esprime la misura in cui il codice software possiede una serie di attributi statici, indipendentemente dall'ambiente di utilizzo e dall'utente
 - ❑ **QUALITA' ESTERNA**: esprime il comportamento dinamico del software, nell'ambito d'uso
- ⇒ **I tre punti di vista si influenzano a vicenda**: non può esservi qualità percepita positivamente dall'utente senza che vi sia una buona qualità intrinseca al codice e buone prestazioni.
- ⇒ Il modello di qualità ISO/IEC 25010 definisce 8 caratteristiche di qualità interne ed esterne:
- ❑ Sicurezza
 - ❑ Affidabilità
 - ❑ Efficienza
 - ❑ Manutenibilità
 - ❑ Portabilità
 - ❑ Funzionalità
 - ❑ Compatibilità
 - ❑ Usabilità



Proprietà di qualità

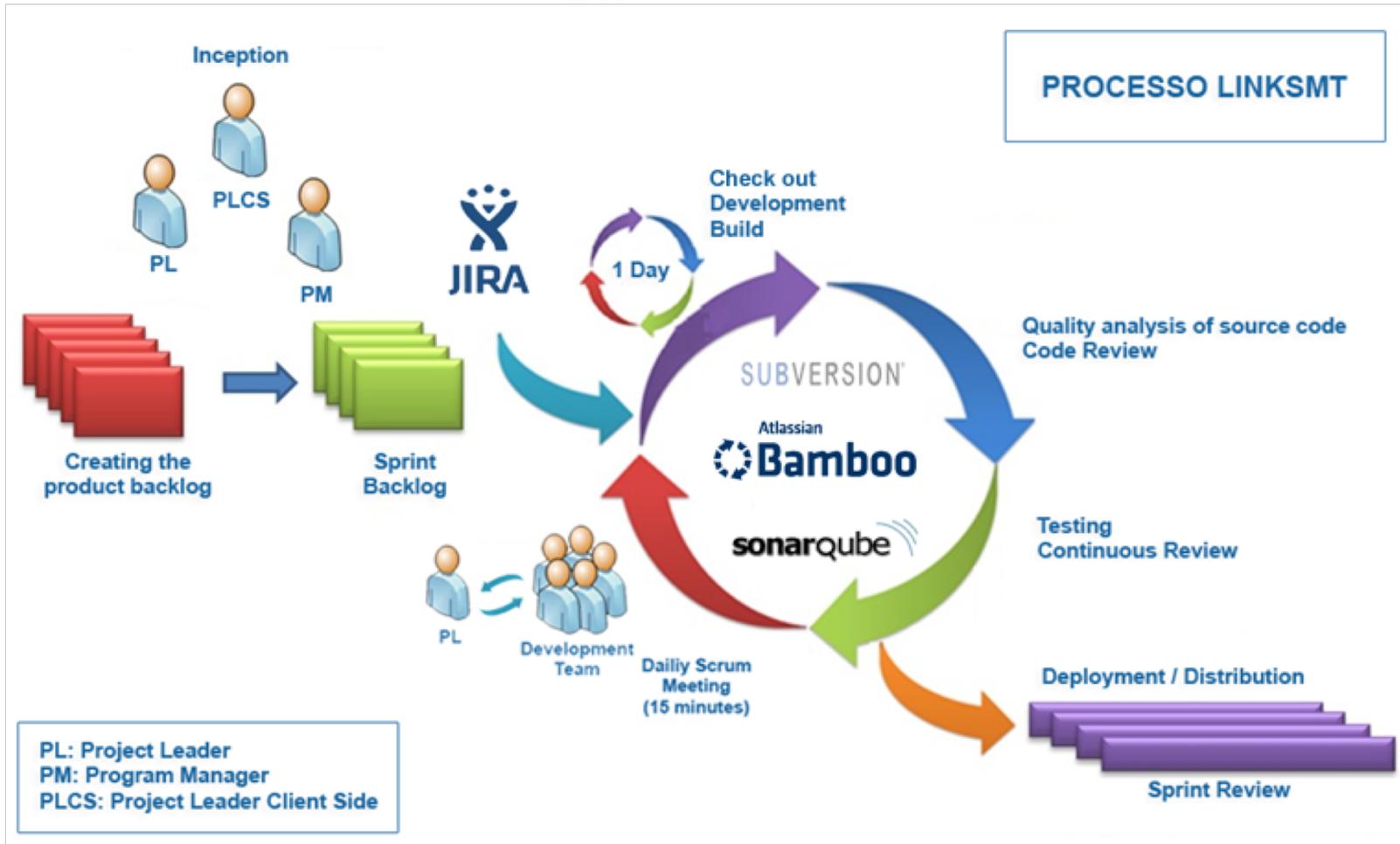


Proprietà di qualità
Documentazione source code
Codice duplicato
Complessità ciclomatica
Strutturazione di classi
Strutturazione di packages
Grandezza delle unità
Bilancio astrazione
Cicli di dipendenza
Violazioni regole analisability
Violazioni regole modularity
Violazioni regole modifiability
Violazioni regole reusability
Violazioni regole testability

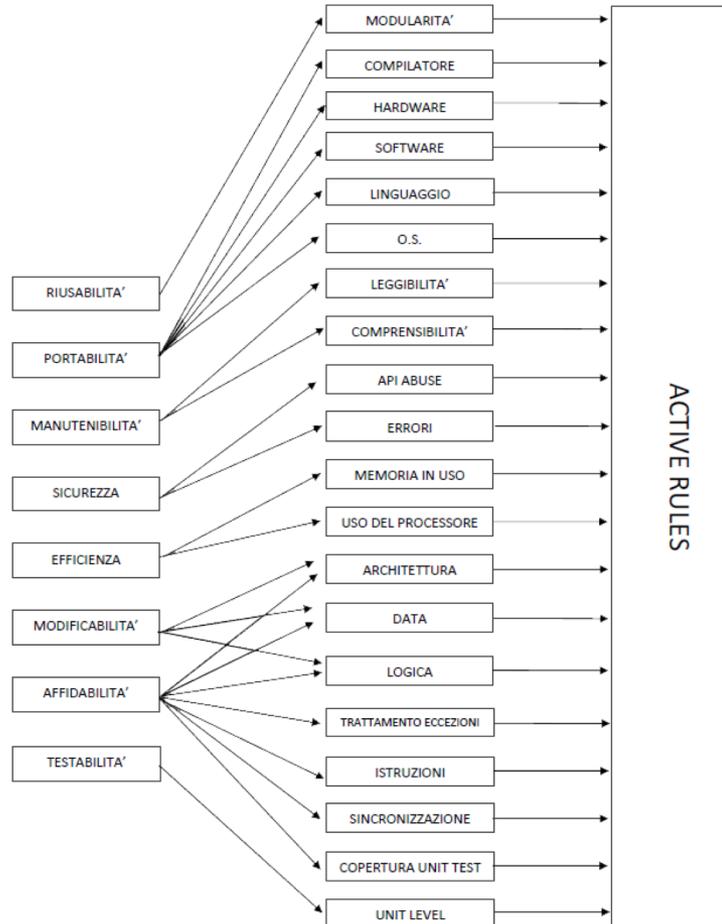
Architettura di supporto

- ⇒ È stata proposta un'architettura a supporto del processo costituita da un set di tools a corredo delle varie fasi:
- ❑ BAMBOO: continuous integration/deployment
 - ❑ JIRA: application lifecycle management
 - ❑ SONARQUBE: quality management
 - ❑ SVN: revision control system

Architettura a supporto del processo



Modello e profilo di qualità SonarQube



Sonar way Java

Rules

245 active rules

21 blocker

70 critical

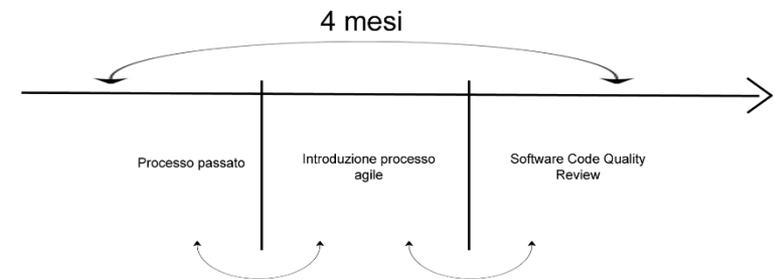
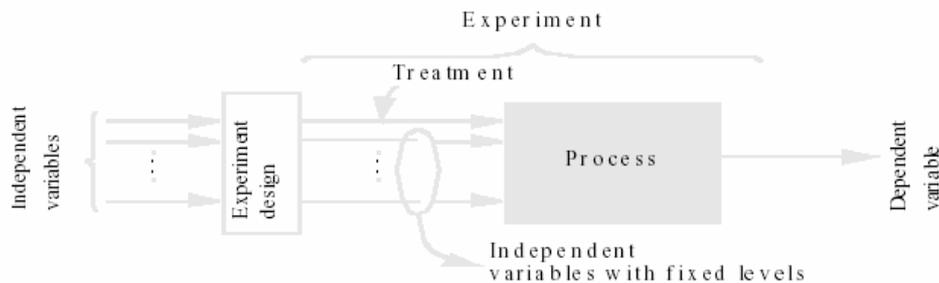
119 major

33 minor

2 info

Sperimentazione in campo

- ⇒ La sperimentazione del processo ha avuto una durata complessiva di 4 mesi ed è stata articolata in tre momenti:
- ❑ Progetto Pilota: Introduzione ed applicazione formale del processo proposto nell'ambito di un progetto industriale denominato «SIDIP»
 - ❑ Miglioramento dell'approccio: avvio della fase di Software Code Quality Review e refactoring del codice sviluppato
 - ❑ Analisi post mortem su progetti sviluppati in passato e comparazione



Le variabili Dipendenti: Debito Tecnico

- ⇒ "Technical Debt (TD) includes those internal things that you choose not to do now, but which will impede future development if left undone. This includes deferred refactoring." *Ward Cunningham's*
- ⇒ In SonarQube il debito tecnico di un prodotto software è misurato in termini di tempo utile al riparo di una certa applicazione
- ⇒ La somma delle stime per tutti quelli che sono i problemi rilevati definisce il debito tecnico totale in giorni-uomo

Analisi dei risultati (1)

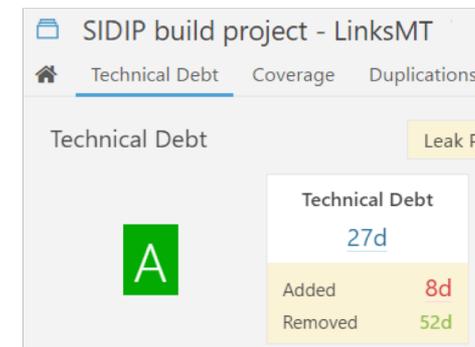
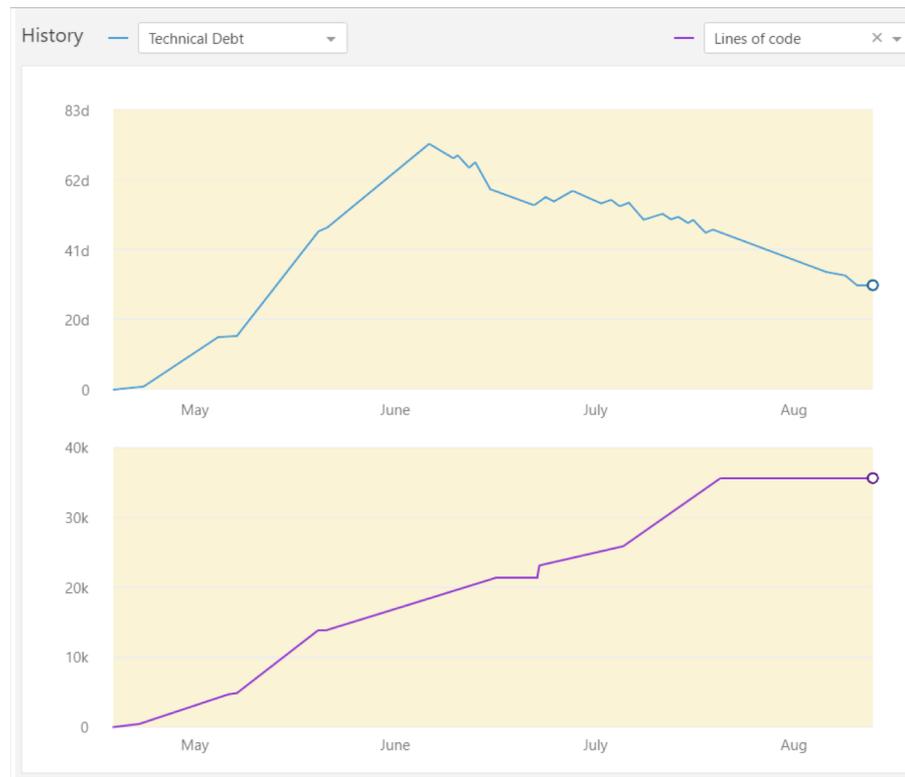
- ⇒ Nei primi 30 giorni di sviluppo si è esclusa volutamente dal processo una fase di Software Code Quality Review al fine di analizzare una situazione dettata da una non gestione della qualità del codice prodotto
- ⇒ I risultati ottenuti da questa prima analisi hanno generato un debito tecnico pari a 79 giorni

CARATTERE	PROBLEMI	PERCENTUALE	STIMA RISOLUZIONE
Blocker	41	1%	1 giorno
Critical	397	9,3%	11 giorni
Major	2294	54%	50 giorni
Minor	1298	30,5%	17 giorni
Info	220	5,2%	10 minuti

- ⇒ Dal 10 Giugno il processo è stato supportato da una fase di Software Code Quality Review:
 - ▣ Si è provveduto ad un refactoring continuo del codice sorgente sviluppato condividendo nel contempo know-how ai membri del team di sviluppo

Analisi dei risultati (2)

⇒ Al 17 Agosto il progetto mostra il più alto valore di qualità (A) avendo proceduto ad un refactoring del codice per un debito tecnico pari a 52 giorni.



Analisi dei risultati (3)

- ⇒ Valutazione e comparazione di due sistemi software prodotti in passato con SIDIP:
 - ❑ TAMPJB (Progetto passato)
 - ❑ MUL (Progetto passato)
 - ❑ SIDIP (Progetto sperimentale)

- ⇒ I software passati hanno adottato un modello di sviluppo differente da quello proposto non includendo una fase di Software Code Quality Review

CARATTERE	MUL	TAMPJB	SIDIP (30 GIORNI)	SIDIP (RISULTATI FINALI)
Blocker	2007	311	41	3
Critical	3236	871	397	69
Major	12865	4495	2294	588
Minor	13970	4845	1298	190
Technical Debt	770d	381d	79d	27d

Analisi dei risultati (4)

	<u>LINKSMT - MUL</u> <u>BUILD</u> 1.0	<u>LINKSMT - TIMPJB</u> <u>BUILD</u> 1.0	<u>SIDIP BUILD PROJECT -</u> <u>LINKSMT</u> 0.1
Lines of code	492130	102580	36037
Complexity	89297	18639	5864
Comments (%)	23.1%	13.4%	5.1%
Duplicated lines (%)	46.1%	35.5%	12.3%
Issues	32453	10587	994
Duplicated blocks	8975	9640	515
Duplicated files	895	351	91
Duplicated lines	420236	57226	5914
Public API	56365	5918	1467
Blocker issues	2007	311	3
Critical issues	3236	871	69
Major issues	12865	4495	588
Minor issues	13970	4845	190
Technical Debt	770d	381d	27d

Conclusioni

- ⇒ Riassumendo, il lavoro di tesi ha focalizzato l'attenzione sulla proposizione di un processo agile orientato alla qualità del prodotto software secondo lo standard ISO/IEC 25000
- ⇒ L'analisi retrospettiva su passati progetti software ha evidenziato un miglioramento significativo della qualità del prodotto software
- ⇒ Il processo proposto è attualmente in uso presso LinksMT



Grazie per l'attenzione

Dipartimento di Informatica - Università degli Studi di Bari
Via Orabona, 4 - 70125 - Bari
Tel: +39.080.5443270 | Fax: +39.080.5442536
serlab.di.uniba.it