



REINGEGNERIZZAZIONE DEL PROCESSO DI MANUTENZIONE IN CERVED

Anno Accademico 2014/2015

Relatore: Danilo Caivano

Laureando: De Cristofaro David

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

Cerved Group s.p.a.



- ⇒ Cerved Group s.p.a. è il più grande *information provider* italiano con sedi sparse su tutto il territorio nazionale.
- ⇒ Valuta la solvibilità e il merito creditizio delle imprese, monitora e gestisce il rischio di credito durante tutte le fasi e definisce strategie di marketing (*Rating finanziario*).

SUPER – Supporto Perizie



- ⇒ Web Service *Java-Based*, istanziato in produzione alla fine del 2012
- ⇒ Caratteristiche Architettureali
 - ❑ L'interoperabilità tra diversi elaboratori su di un contesto distribuito (SOA - prerogativa base di un WS)
 - Supporto WSDL, Comunicazione SOAP
 - ❑ Disaccoppiamento tra la logica di sistema e l'interfaccia utente
 - STRUTS 2 (*Pattern MVC*)
 - ❑ Gestione della persistenza dei dati
 - ORM Hibernate
- ⇒ Il team di sviluppo e di manutenzione è formato da quattro sviluppatori e il team manager

SUPER – Supporto Perizie



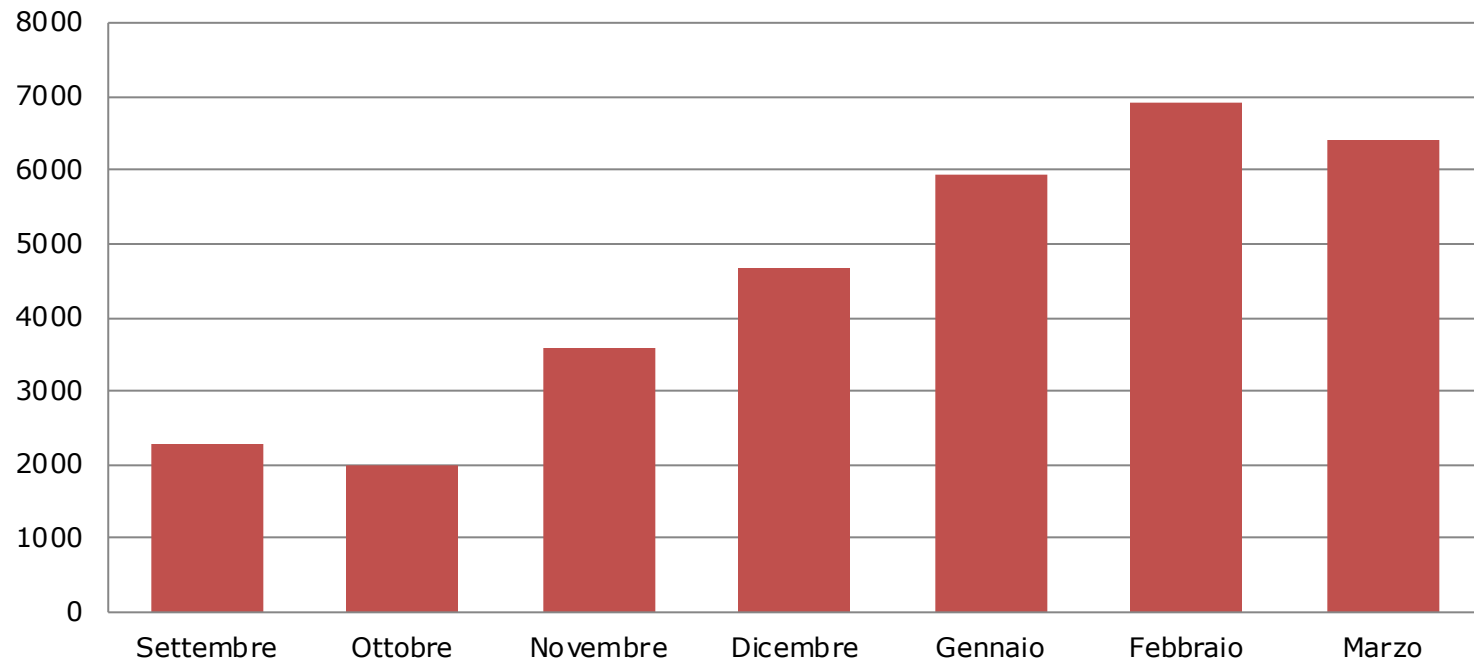
⇒ Caratteristiche Funzionali

- ❑ Gestione del ciclo di vita della perizia
 - ❑ Assegnazione automatica della perizia
 - ❑ *Mail Sender*
 - ❑ Controllo del traffico delle perizie, applicando determinati filtri di ricerca
 - ❑ Gestione (inserimento, modifica e cancellazione) degli utenti, clienti e dei modelli delle perizie
 - ❑ Monitoraggio dei flussi di comunicazione con i servizi comunicanti col sistema
 - ❑ Generazione di report riepilogativi, in formato .xlsx
- ⇒ Vengono gestiti, inoltre, diversi livelli di utenza che possono accedere a determinate funzionalità

Scopo dello Studio – Analisi dello scenario

⇒ Per la durata di 6-7 mesi di studio sul campo, è stato approfondito il processo di manutenzione di SUPER e, contemporaneamente, si è avuto un forte incremento delle pratiche gestite nel sistema.

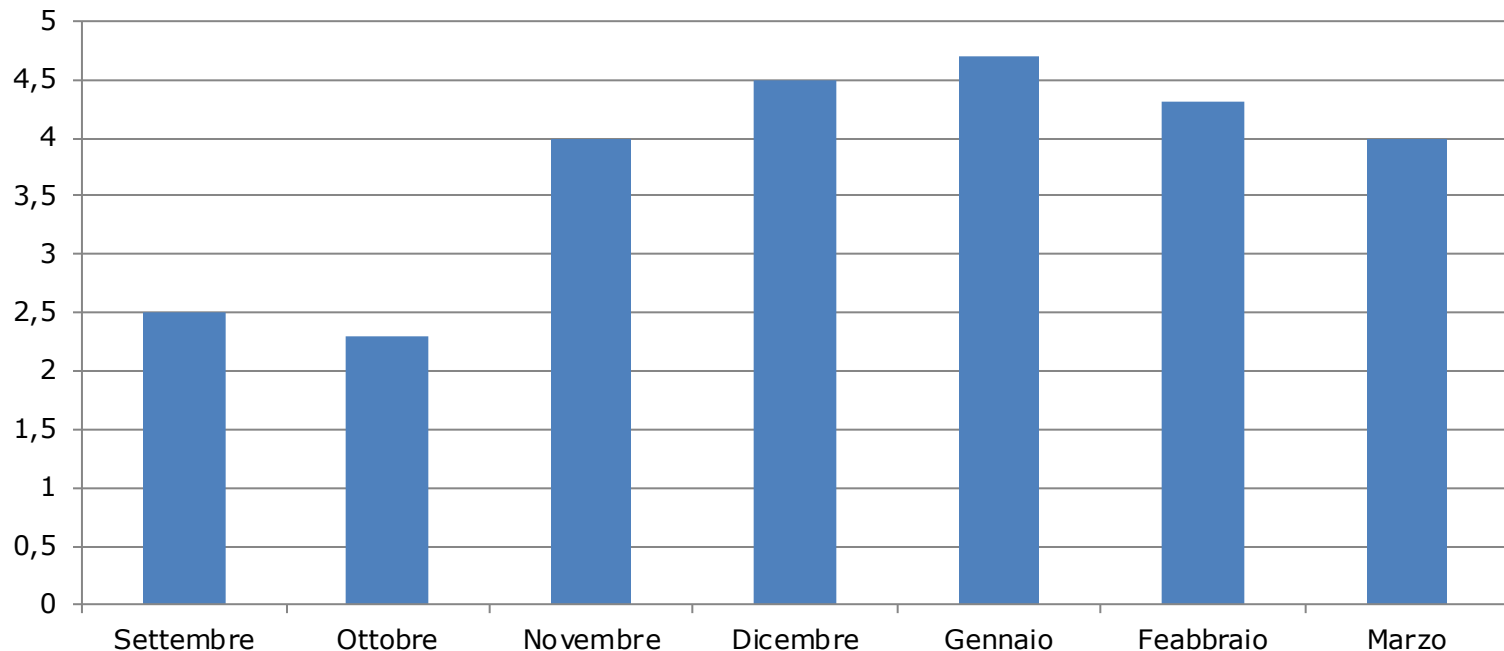
Pratiche Completate in SUPER



Scopo dello Studio – Analisi dello scenario

⇒ Un aumento di pratiche gestite corrisponde un aumento di utenti attivi che gestiscono le stesse e, di conseguenza, la probabilità di affioramento di anomalie del sistema incrementa.

Segnalazioni Ricevute dall'IT in una settimana



Scopo dello Studio – Analisi dello scenario

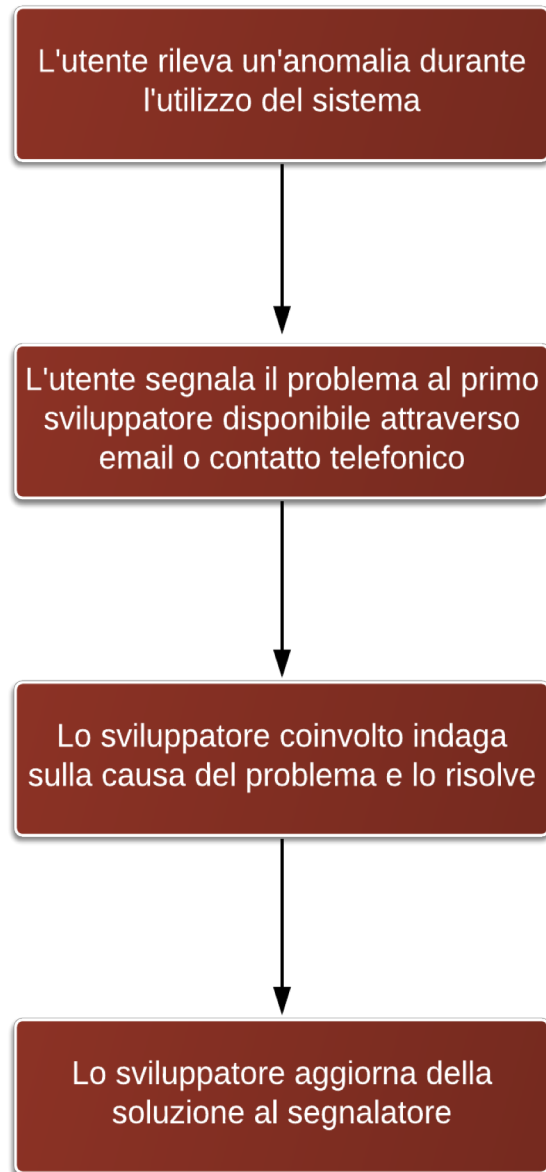
- ⇒ Queste ultime segnalazioni sono riferite a bug funzionali e a necessità di nuove funzionalità che differiscono da implementazioni dovute da nuovi accordi commerciali (non misurabili nel lasso di tempo).
- ⇒ Si conferma la necessità di REINGEGNERIZZARE l'attuale *processo di manutenzione*.
- ⇒ I benefici che ne scaturiscono sono:
 - ❑ Ottimizzazione delle risorse dispiegate
 - ❑ Risparmio in termini di tempo
 - ❑ Aumento dell'efficienza (conseguenza dei benefici espressi precedentemente)
- ⇒ Manutenzione del Software:
 - ❑ Tutte quelle attività che definiscono i processi correttivi ed evolutivi che avvengono dopo il rilascio del sistema in ambiente di produzione



Scopo dello Studio

Analisi dello scenario

- Sia attività correttive che evolutive sono affrontate dal team di sviluppo in maniera del tutto *informale*.
- Punti di debolezza per le attività correttive:
 - Spreco di tempo (contatto tra utente sviluppatore, identificazione del problema, attività di fixaggio)
 - Mancanza di gestione del lavoro
 - Mancanza di intercomunicabilità tra i membri del team
 - Mancanza di un quadro rappresentativo che dimostri al segnalatore l'avanzamento della sua richiesta in tempo reale



Scopo dello Studio

Analisi dello scenario

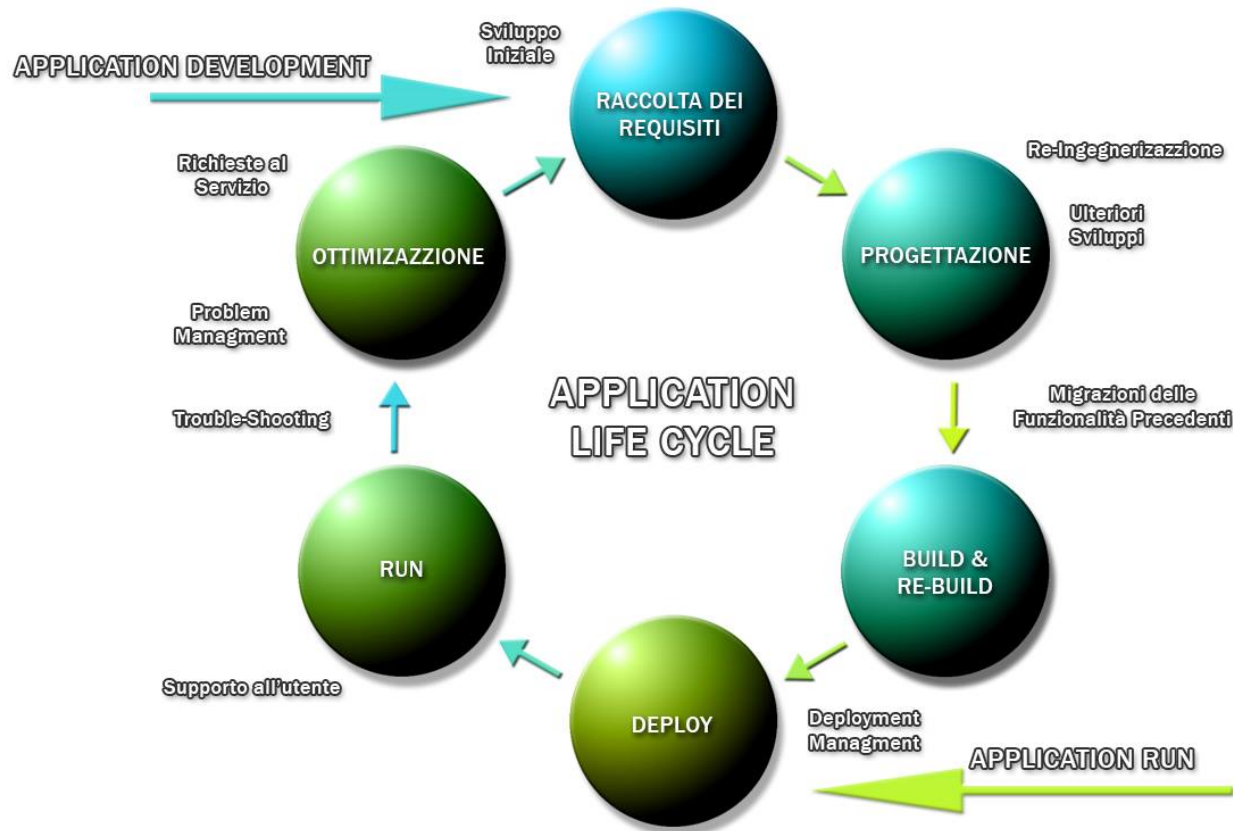
- Punti di debolezza di attività evolutive:
 - La natura lineare del modello porta a stati "bloccanti", nei quali alcuni membri del team devono attendere che i loro colleghi completino una certa attività
 - Non si ha un riscontro graduale con le aspettative del cliente per capire se l'implementazione procede nella direzione giusta o no
 - Si rischia di mantenere o, nel peggiorare delle ipotesi, di abbassare il livello di qualità complessiva delle funzionalità offerte e, quindi, dell'intero sistema.



Approccio Proposto

- ⇒ Un processo efficace di *Application Management* per Super può:
- ❑ Aumentare la produttività (conseguire un risultato superiore ai mezzi impiegati)
 - ❑ Migliorare la qualità complessiva del sistema
 - ❑ Azzerare le barriere di comunicazione tra gli *stakeholders*
 - ❑ Accelerare lo sviluppo software
 - ❑ Dimezzare i tempi di manutenzione
 - ❑ Massimizzare l'investimento sulle capacità, sui processi e sulle tecnologie
 - ❑ Incrementare la flessibilità, riducendo il tempo ed i costi di integrazione con nuove funzionalità

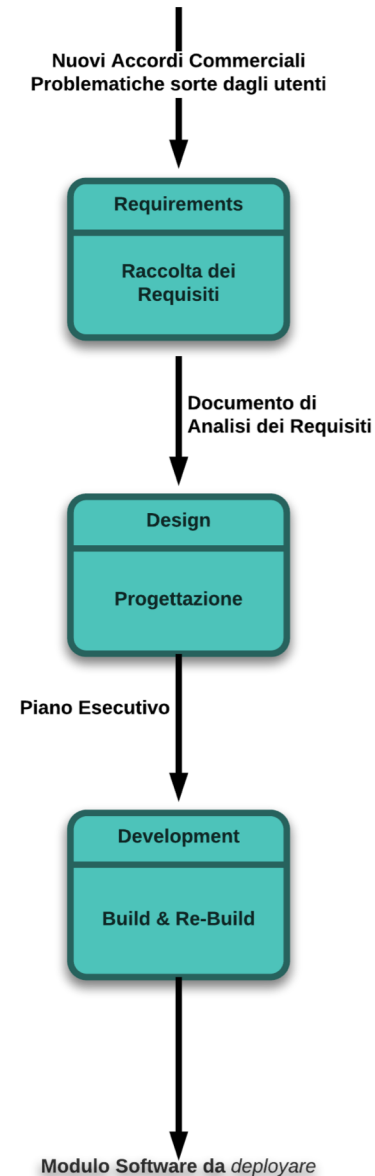
Approccio Proposto – Descrizione del Processo



- ⇒ Il punto focale di questo modello è la combinazione armonica di tutte le attività coinvolte sia nello sviluppo di nuove funzionalità sia nel fixaggio di malfunzionamenti del sistema durante tutto il ciclo di vita del sistema (correzione & evoluzione).
- ⇒ Nello studio di tesi è stato usato il formalismo *FSP*, con la rappresentazione dei manufatti, dei WFD, degli scenari procedurali e dei tool adottati.

Application Development

- Questa fase racchiude tutte le attività del processo che costruiscono tutta l'impalcatura per lo sviluppo di moduli integranti l'applicativo.
- Il processo *parte* nel momento in cui sorgono necessità e problematiche che richiedono un nuova integrazione, che intanto continua a *girare* sul server di produzione.



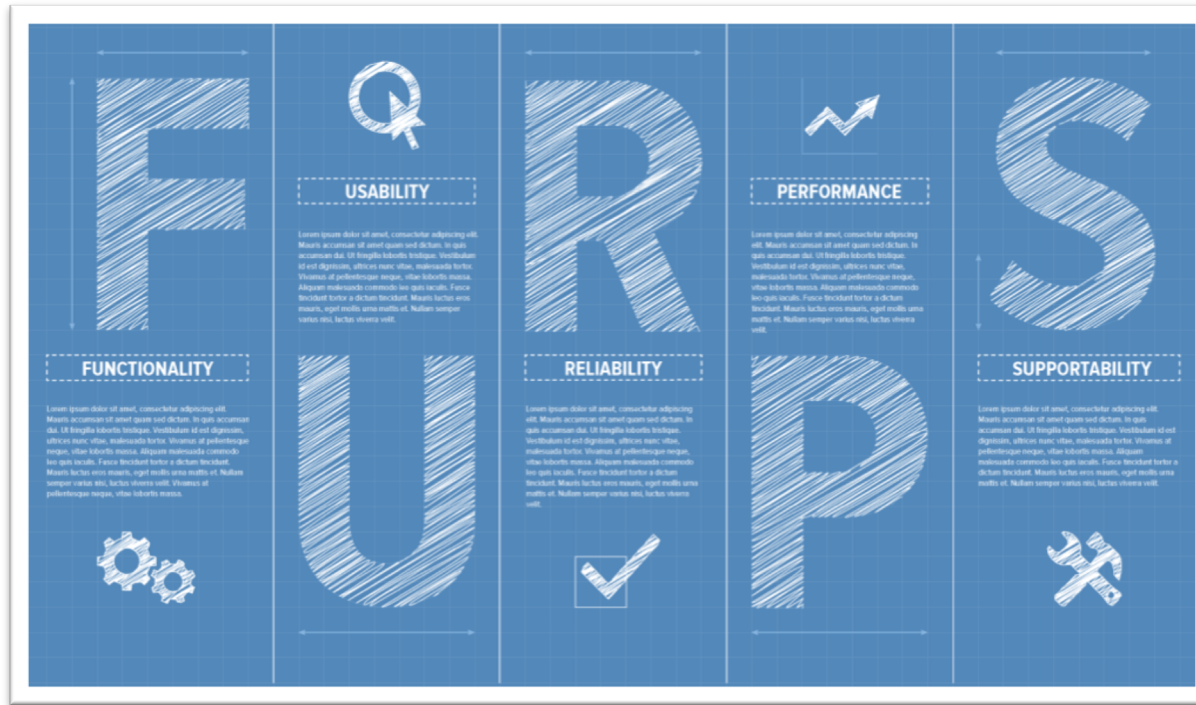
Application Development

Analisi dei Requisiti

- Il team di sviluppo comincia un lavoro di ELICITAZIONE, ricavando informalmente informazioni che rispecchiano le aspettative effettive del richiedente
- Successivamente, si passa ad un'ANALISI dei dati raccolti per stabilirne la fattibilità ed ad una NEGOZIAZIONE con il richiedente
- Si passa poi alla SPECIFICA DEI REQUISITI in cui si analizzano gli aspetti che concretizzano, dal punto di vista implementativo, i dati raccolti.
- Nella fase di VERIFICA & VALIDAZIONE, viene messo a confronto con lo standard FURPS+ che risalta aspetti importanti per la coerente integrazione da realizzare.



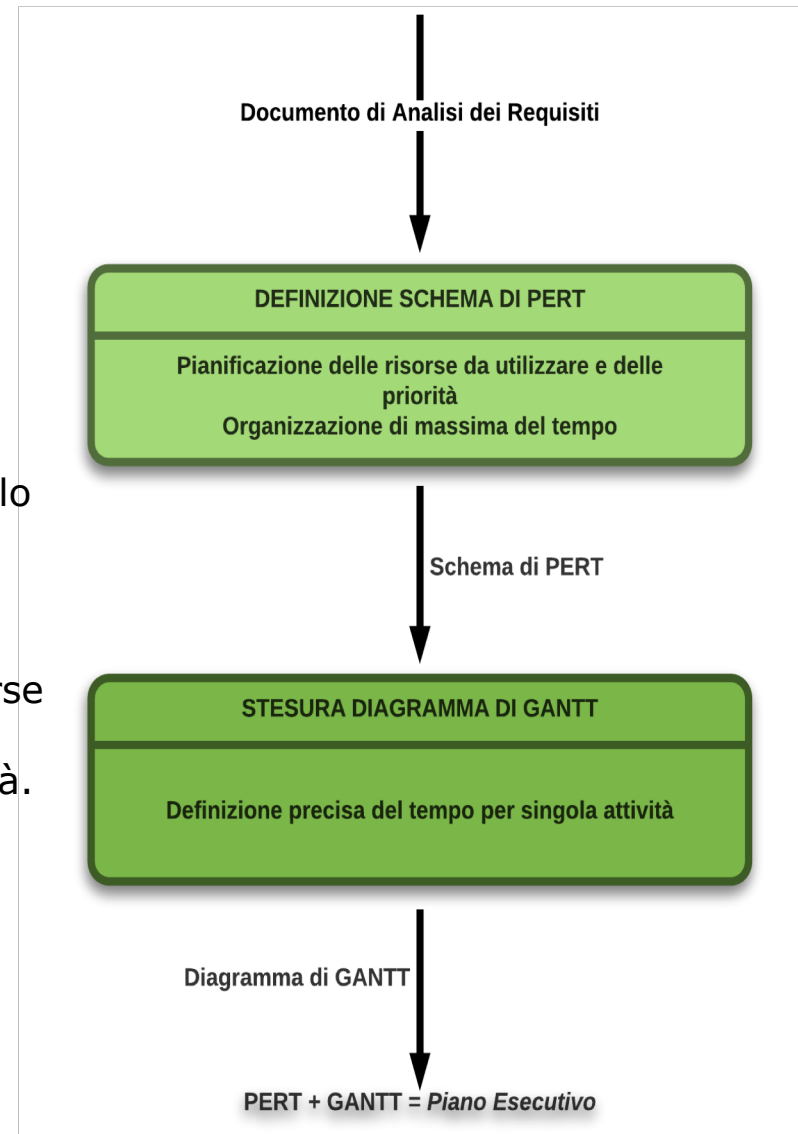
Application Development - Analisi dei Requisiti



- ⇒ Ideato da Robert Grady, FURPS+ comprende anche caratteristiche secondarie non prese in considerazione dalla prima versione del framework.
- ⇒ Una volta che sono state confrontate le specifiche software con FURPS, viene generato il documento che riassume i requisiti e che possono essere trattati per la progettazione dell'implementazione.

Application Development Progettazione

- Generato il documento di analisi dei requisiti con aspetti concreti da realizzare, si passa alla fase di *progettazione* che delinea le attività da svolgere con priorità, risorse e tempistiche da assegnare.
- Viene prodotto lo SCHEMA DI PERT, delineando ordine delle attività, priorità e risorse con un livello di astrazione dettagliato
- Successivamente, si stende il DIAGRAMMA DI GANTT nel quale si assegnano le attività alle risorse nello specifico e si definisce nel dettaglio il tempo necessario per lo svolgimento delle singole attività.
- Risultato dell'unione di entrambi i lavori è il PIANO ESECUTIVO



Application Development - Progettazione

Schema di PERT

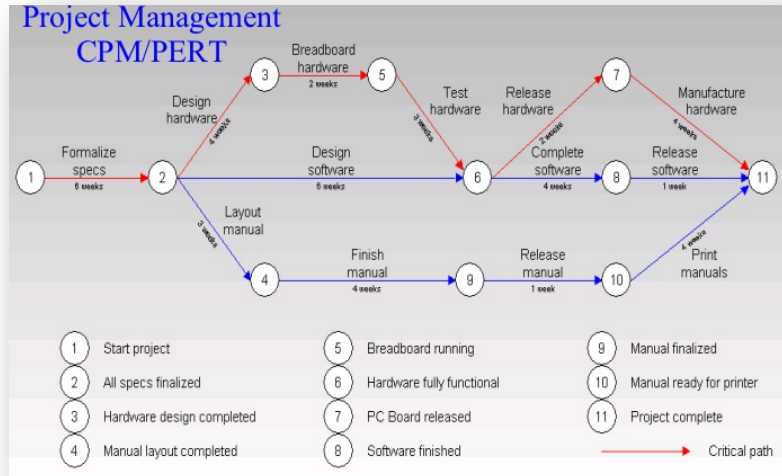
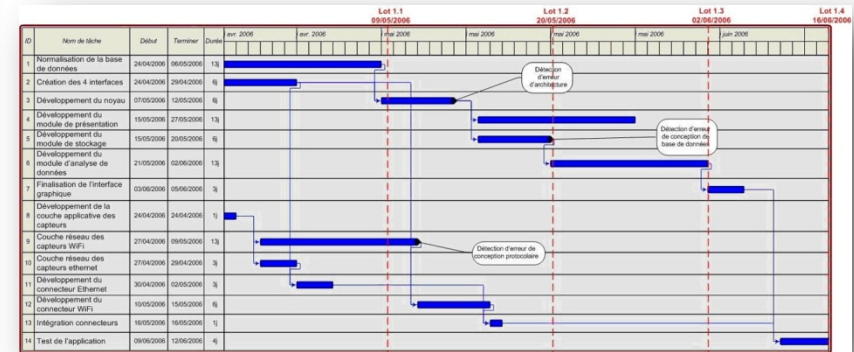


Diagramma di GANTT

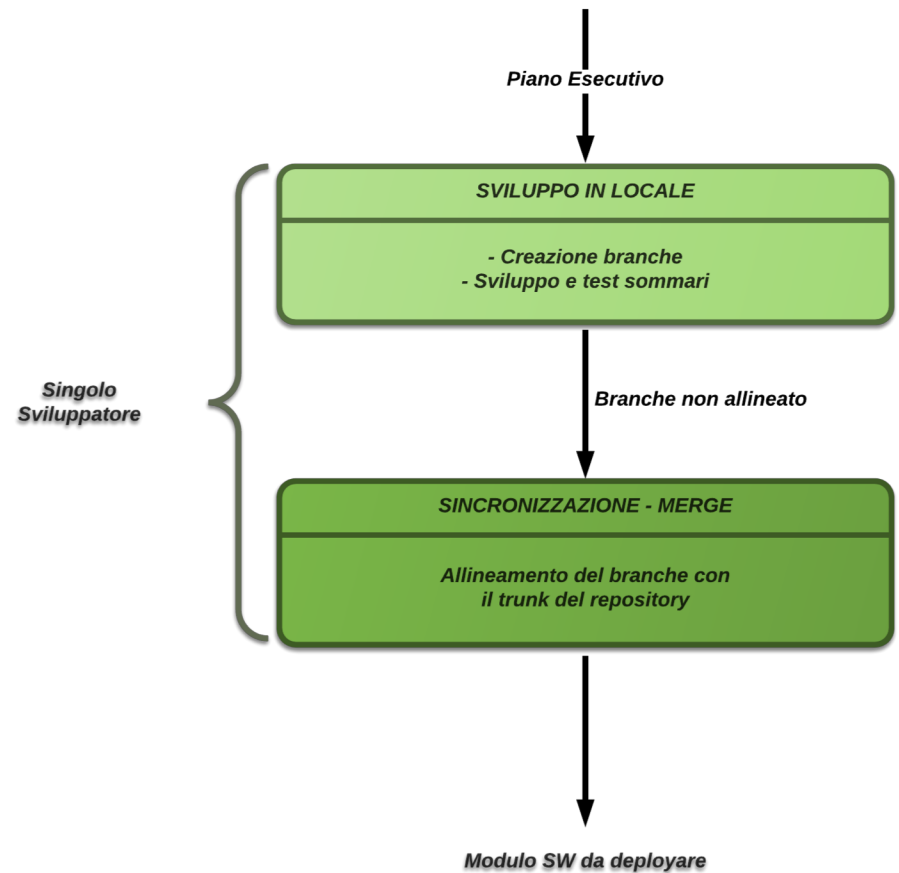


- Il *PERT* (*three-point-estimation*) è un metodo statistico di determinazione dei dettagli delle attività di progetto. Rispetto alla semplice stima a valore singolo, il metodo presuppone la determinazione di valori di stima ottimale, probabile e pessimistico che risultano più adeguati a valutare tempi e costi di attività di progetto che presentano incertezza o complessità.

- Il *diagramma di Gant* è costruito partendo da un asse orizzontale (arco temporale totale del progetto) e da un asse verticale (attività che costituiscono il progetto). Permette la rappresentazione grafica di un calendario di attività, utile al fine di pianificare, coordinare e tracciare specifiche attività in un progetto dando una chiara illustrazione dello stato d'avanzamento del progetto rappresentato.

Application Development Build & Re-Build

- Prima dare inizio allo SVILUPPO IN LOCALE, si effettua una copia del sistema preposta allo scopo del processo (branche), prelevata dalla versione più recente del sw in produzione (trunk)
- Il branche non allineato è pronto per la SINCRONIZZAZIONE (o MERGE) con il trunk.
- Pratica considerata nel AM è la *Continuous Integration* che avviene attraverso un sistema di *versioning*; consiste nell'allineamento frequente dagli ambienti di lavoro degli sviluppatori verso l'ambiente condiviso (*svn repository*)
- Il *versioning* del sw è gestito dagli sviluppatori grazie al tool di supporto *Tuleap*.



Application Development Build & Re-Build



⇒ La Enalean, azienda francese che opera nell'automazione dei processi di sviluppo sw, ha rilasciato Tuleap, web browser application concepita per l'AM. Le caratteristiche che interessano il nostro processo sono:

- ❑ Fornisce una differenza visuale nel confronto tra le versioni e permette la ricerca dettagliata tra le patch
- ❑ Migliora la tracciabilità, forzando l'inserimento dei messaggi al momento del commit
- ❑ Assegna determinati diritti di accesso o di scrittura a determinate directory dei branches agli utenti del repository

asprintf.c - Browse SVN Tree

Subversion Info Browse SVN Tree My SVN Commits SVN Query SVN Admin Email Notification Help

[V-Model] / asprintf.c

Diff of /asprintf.c

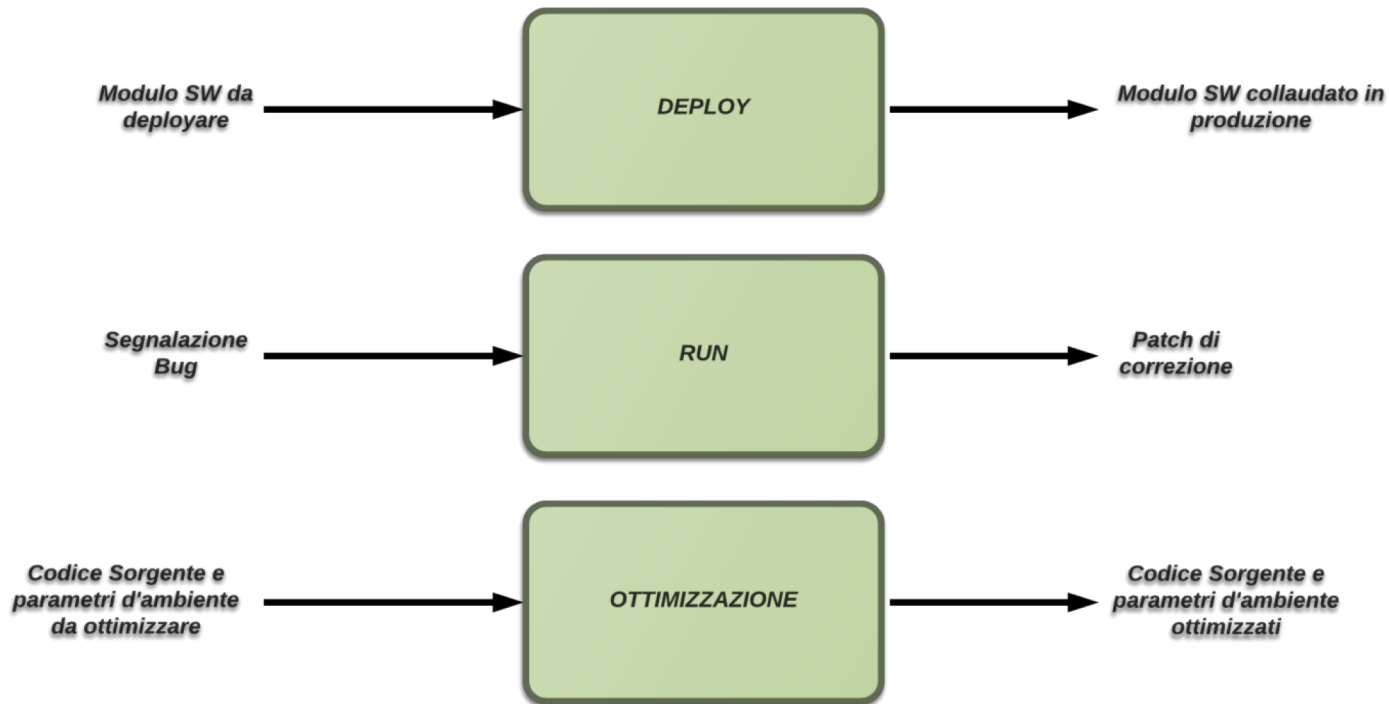
Parent Directory | Revision Log | Patch

#	Line 19	Line 19
19	not, write to the Free Software Foundation, Inc., 51 Franklin Street - Fifth Floor, Boston, MA 02110-1301, USA. ʘ	not, write to the Free Software Foundation, Inc., 51 Franklin Street - Fifth Floor, Boston, MA 02110-1301, USA. ʘ
20		
21	#ifdef HAVE_CONFIG_H	
	#include "config.h"	
	#endif	
	#include "ansidecl.h"	
	#include "liberty.h"	
	#include <stdarg.h>	
22		
23		
24		
#	Line 47	Line 40
40	asprintf (char **buf, const char *fmt, ...)	asprintf (char **buf, const char *fmt, ...)
41	{	{
42	int status;	int status;
	VA_OPEN (ap, fmt);	
43	VA_FIXEDARG (ap, char **, buf);	VA_FIXEDARG (ap, char **, buf);
44	VA_FIXEDARG (ap, const char *, fmt);	VA_FIXEDARG (ap, const char *, fmt);
45	status = vasprintf (buf, fmt, ap);	status = vasprintf (buf, fmt, ap);

Colored Diff Legend:
 Removed from v.5
 changed lines
 Added in v.12

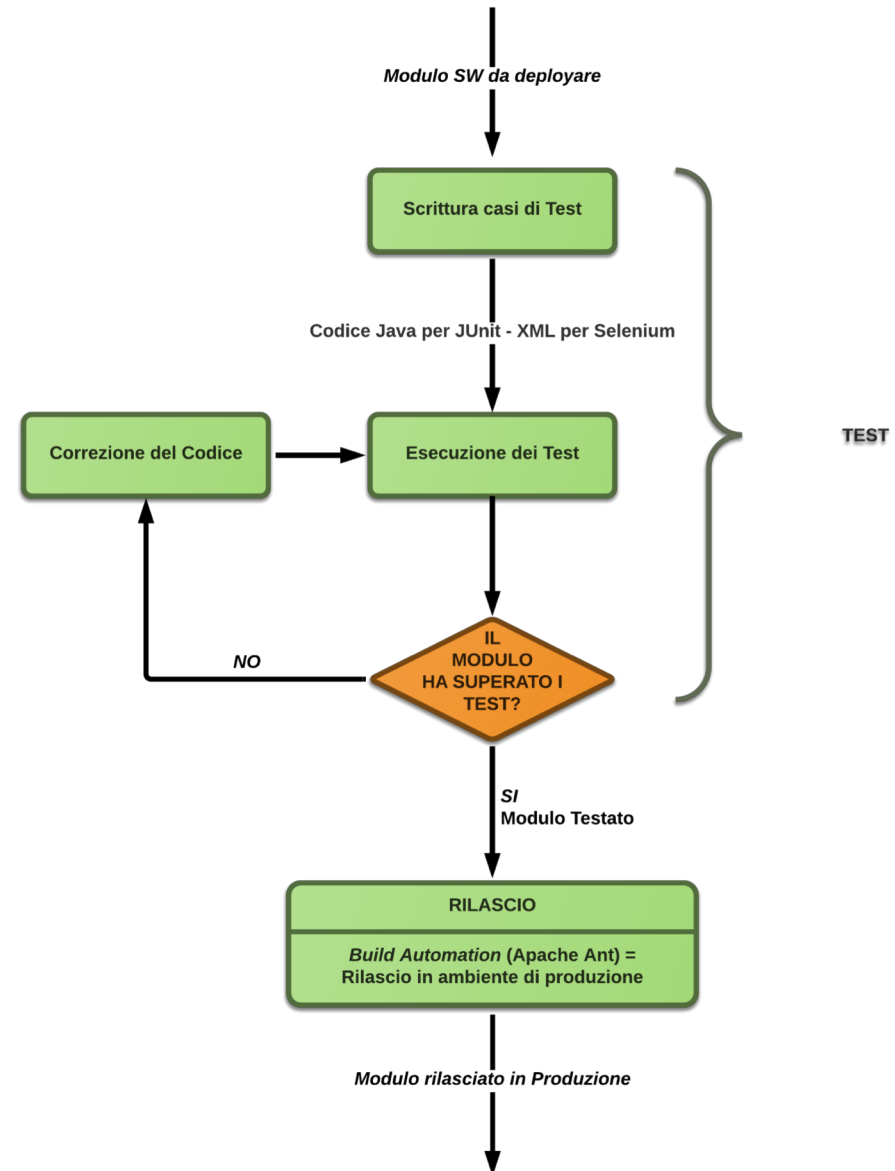
Application Run

- ⇒ Il passaggio dalla macrofase di APPLICATION DEVELOPMENT a quella di APPLICATION RUN è praticamente immediata.
- ⇒ Si noti che le tre fasi di Application Run sono volutamente svincolate fra loro proprio perché è possibile che assumano un ordine diverso da quello del modello.



Application Run Deploy

- Nella parte di TEST, il modulo sw è sottoposto ad una serie di test sia lato logico (JUnit) sia lato client (Selenium)
- Il modulo, superato positivamente tutti i casi di test preposti, passa alla fase di RILASCIO, in cui il sw deployato viene "dato in pasto" ad un tool per il *build automation* (Apache Ant) che permette di astrarre dai dettagli di rilascio, gestendolo in maniera più sicura e controllata e, perciò, rilasciando la nuova versione sw in produzione.



Application Run - Deploy (Test)

- ⇒ Ogni integrazione ingente ha un impatto sia a livello server che a livello client; ciò necessita un doppio livello di astrazione nell'esecuzione dei test.
- ⇒ *JUnit* è un tool per l'automazione del testing di moduli Java che si ispira alla tecnica del *Test Driven Development*. Ogni test contiene le asserzioni che controllano che i moduli testati non contengano difetti, e vengono "scritti" allo stesso livello di implementazione in cui sono stesi i moduli software.

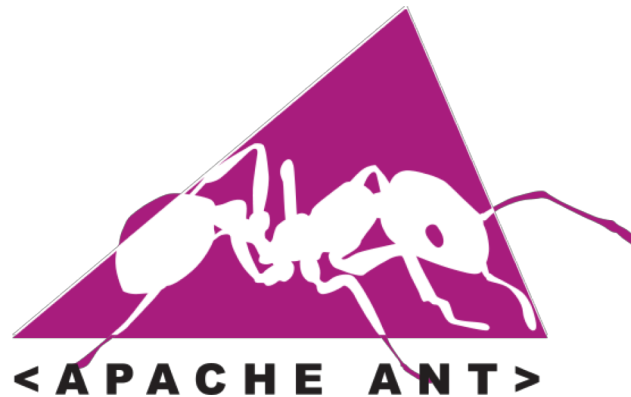


JUnit

- ⇒ *SELENIUM* è uno strumento Open Source di software testing per applicazioni web che permette di registrare ed eseguire test in maniera automatica, astraendo dal dettaglio di esecuzione manuale.
- ⇒ Questo tool, inoltre, è multiplatforma: grazie alla stesura e alla esecuzione di *macro*, le interfacce sono testate in maniera *user-like*, simulando l'utilizzo delle funzionalità via interfaccia da parte di un utente.

Application Run - Deploy (Release)

- ⇒ Una delle priorità è il poter gestire *configuration, change e release management* insieme, avendo una funzione centralizzata per gestirli.
- ⇒ Apache Ant è un progetto Apache, open source, ed è un build tool, utilizzato per la compilazione, e l'installazione automatica di applicazioni Java complesse, come *Enterprise Application*.

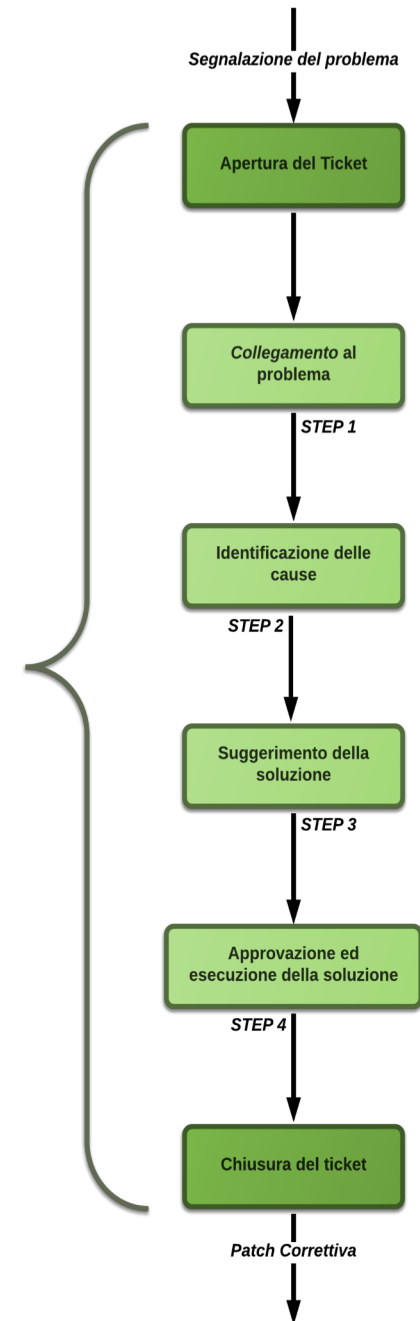


- ⇒ Ant esegue i suoi compiti, seguendo le istruzioni definite in un file chiamato *build.xml*, in cui sono presenti speciali tag i quali servono a definire i task da elaborare.
- ⇒ Essendo per sua natura estendibile, è praticamente possibile gestire qualsiasi compito, aggiungendo nuove librerie.

Application Run Run Support - TTS

- Nella fase *run*, viene curato il supporto all'utente finale che può trovare un o non ha le conoscenze necessarie del sistema per proseguire nel proprio lavoro.
- L'intero iter è definito in un *Trouble Ticket System (TTS)*
- Tutte le successive fasi sono implementate dal tool *JIRA Service Desk*.

TROUBLE
TICKET
SYSTEM



Application Run - Run Support - TTS

JIRA Service Desk

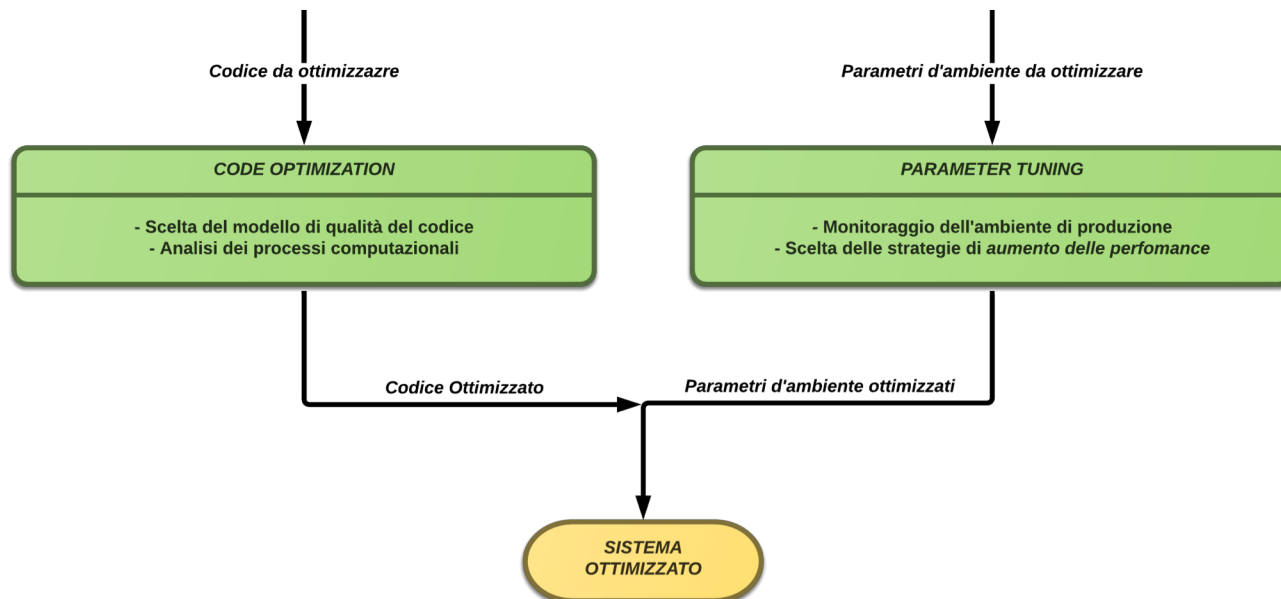
⇒ Prodotto da Atlassian, questo tool non fornisce solamente un sistema di *ticketing*, ma ha diverse caratteristiche che lo rendono estremamente funzionale come:

- ❑ Memorizzazione dei ticket gestiti e suggerimenti automatizzati basati sulla conoscenza
- ❑ Automazione di gestione del carico di lavoro ed esecuzione automatica di task "ripetitivi"
- ❑ Supporta le *best practice* suggerite dal modello ITIL proprie di questa fase



Application Run - Ottimizzazione

- ⇒ Nella fase di *ottimizzazione* si analizzano le logiche coinvolte nelle funzionalità e i parametri server per migliorare le prestazioni del sistema nell'insieme.
- ⇒ L'attività di CODE OPTIMIZATION delinea un determinato modello di qualità sul quale comparare il codice sorgente prodotto e controllare l'efficienza degli algoritmi (*Kiwuan*)
- ⇒ Il PARAMETER TUNING analizza nel dettaglio i parametri delle macchine virtuali su server, attraverso una piattaforma APM (AppDynamics), che vengono modificati in base alle criticità rilevate.



Application Run – Ottimizzazione – Code Optimization

⇒ Kiuwan è un *SaaS (Software as a service)* ed è finalizzato all'analisi *statica* del programma per la gestione dei processi di sviluppo, della qualità e della sicurezza del software. Fornisce:

- ❑ Misurazione e analisi immediata e continuata del codice
- ❑ Piattaforma cloud per l'applicazione
- ❑ Customizzazione del modello di qualità da adottare
- ❑ Piattaforma che supporta il modello decisionale
- ❑ Indicazioni pratiche e immediate delle soluzioni da adottare per il codice

kiuwan
DIVE IN YOUR CODE



Application Run – Ottimizzazione – Parameter Tuning

APPDYNAMICS

- ⇒ *AppDynamics*, società leader nel settore dell'*Application Performance Management (APM)*, offre un *Application Intelligence Platform*, accessibile via browser, e che fornisce diversi livelli di monitoraggio come:
- ❑ Real-Time delle statistiche di sistema (l'esperienza del cliente e risultati delle operazioni e dei processi di business – *Application Analytics*)
 - ❑ Synthetic Monitoring (calcolo proattivo delle performance e di visibilità del sistema)
 - ❑ User Experience (esperienza utente browser-based)
 - ❑ DB applicativo
 - ❑ Server sui quali è distribuita l'applicazione

Sperimentazione

- ⇒ Parte di questa reingegnerizzazione è stata già applicata per circa un 20%:
 - ❑ (BUILD & RE-BUILD 100% - DEPLOY 100% - OTTIMIZZAZIONE 40%)
 - ❑ BUILD & RE-BUILD, totalmente integrata - 100%
 - ❑ DEPLOY, totalmente integrate - 100%
 - ❑ OTTIMIZZAZIONE, AppDynamics operativo sul server di produzione, analisi sui dati in corso - 40%
- ⇒ Durante la fase di studio, dato l'incremento delle richieste per l'entrata di nuovi clienti, tutte le risorse del team sono state investite per il raggiungimento, nei tempi previsti, degli obiettivi definiti dagli accordi commerciali.



Conclusioni

- ⇒ L'importanza di questo studio e la forte necessità di reingegnerizzazione fornisce alta priorità all'applicazione, in tempi brevi, di questo modello.
- ⇒ Non vengono presi in considerazione altri aspetti positivi che scaturiscono da un aumento di efficienza di questo processo di manutenzione ma che risultano molto rilevanti per una comparazione con altri provider che emettono un servizio simile, dando all'immagine di Cerved maggiore professionalità, affidabilità e qualità.





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