



**DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA DEI SISTEMI DEL TERRITORIO E  
DELLE COSTRUZIONI**

**RELAZIONE PER IL CONSEGUIMENTO DELLA LAUREA MAGISTRALE IN INGEGNERIA  
GESTIONALE**

***Evolution of a web software: from requirements analysis  
to production release***

**SINTESI**

---

RELATORI

IL CANDIDATO

Prof. Riccardo Dulmin

*Dipartimento di Ingegneria dell'Energia,  
dei sistemi, del Territorio e delle costruzioni*

Francesco Arlia

*francescoarlia@hotmail.it*

Dott. Giovanni Iovine

*Technology Consulting Manager*

Sessione di Laurea Magistrale del 27/11/2019

# **Evolutive di un software web: dall'analisi dei requisiti al rilascio in produzione**

**Francesco Arlia**

---

## **Sommario**

L'elaborato di tesi è basato su un'esperienza di tirocinio della durata di tre mesi presso la società di consulenza Accenture S.p.A., nella divisione Technology Consulting Product. L'esperienza è stata svolta all'interno di un team di progetto che ha come scopo la realizzazione di evolutive di un software web per una società operante nel settore dei trasporti sul territorio nazionale. Durante il periodo di attività, è stato possibile conoscere e prendere parte alle principali attività di analisi e sviluppo dei requisiti web, implementati sui sistemi informativi già in essere della società.

Dunque, l'obiettivo principale dell'elaborato è quello di delineare e descrivere il processo di pianificazione e implementazione delle attività eseguite durante il periodo di svolgimento del progetto, focalizzando l'attenzione, in ogni fase del processo, sulle risorse, l'effort necessario, gli output prodotti, i rischi, i ruoli e le responsabilità. Vengono prese in considerazione tutte le fasi: la pianificazione, l'analisi dei requisiti, il disegno, la fase di sviluppo, di test e del rilascio in produzione dei requisiti oggetto del progetto.

## **Abstract**

This thesis work is based on a three months internship to Accenture S.p.A., in Technology Consulting Product. The purpose of the project team where the work has been done, is the implementation of an evolutive software web for a transport sector society. During this period, it was possible to learn the principles activities in web analysis and development processes.

The purpose of this thesis work is to describe the planning processes and tasks implementation with a particular attention to resources, efforts, outputs, risks, rules and responsibilities. The following steps analysed: planning, requirements analysis, development, test e release in production.

## **Ambito e obiettivo**

La presente relazione è basata su un'esperienza di tirocinio curriculare della durata di tre mesi presso la società di consulenza Accenture S.p.A, grazie alla quale è stato possibile entrare a far parte di un team di progetto il cui scopo è quello di definire, pianificare e realizzare evolutive di un software web per un'azienda di trasporti operante sul territorio nazionale. Le mansioni svolte hanno rappresentato un primo approccio verso la figura dell'analista funzionale operante nel settore dell'information technology. Nello specifico le attività eseguite sono state: la stesura delle specifiche funzionali per i sistemi di front end, la scrittura del testbook inerente i requisiti del progetto, l'esecuzione degli scenari di test, il tracciamento delle anomalie applicative e il test di collaudo con l'utente (UAT). L'obiettivo dell'elaborato è quello di delineare le possibili metodologie d'approccio per lo sviluppo software e la definizione delle attività svolte per la realizzazione delle evolutive di progetto.

## **1. Metodologie di sviluppo software a confronto**

### **1.1 Introduzione**

Un software è il prodotto intangibile di un sistema elettronico ed è il risultato di un processo strutturato di sviluppo. Tale processo è costituito da una serie di attività, eventi, documenti e procedure necessarie per la realizzazione dello stesso.

### **1.2 Le metodologie di sviluppo**

L'ingegneria del software è la disciplina che definisce le fasi del ciclo di vita del software e le possibili metodologie di sviluppo. La metodologia di sviluppo software ha il compito di scomporre le attività del processo di sviluppo in sotto-attività coordinate al fine suddividere le fasi e ottimizzare la realizzazione dello stesso. Le metodologie di sviluppo software si dividono in:

- Metodologia Waterfall;
- Metodologia Agile.

#### **1.2.1. La metodologia Waterfall**

Secondo questo modello, il processo di realizzazione del software è strutturato in una sequenza lineare di fasi o passi non sovrapponibili. Si tratta di un approccio lineare e sistematico che deve il suo nome a *cascata* al fatto che il modello si sviluppi in modo tale che l'output di una fase, che prende il nome di deliverable, sia l'input di quella successiva e così a seguire. Le fasi sono le seguenti:

- Studio di fattibilità: ha lo scopo di determinare da un punto di vista tecnico ed economico, se sia possibile realizzare il sistema del quale si vanno definendo i requisiti;

- **Analisi dei requisiti:** in questa fase viene definito l'insieme tutte le funzionalità di cui il software dovrà disporre, progettato e testato un concept grafico che tenga conto delle funzionalità richieste, generato un documento di analisi funzionale che descriva in maniera chiara e dettagliata ogni funzionalità e possibile.
- **System Design:** lo scopo di questa fase è quello di produrre dei documenti che definiscano quelli che saranno i componenti (moduli) che andranno a costituire il sistema e la sua architettura (ad esempio i database, i server, le librerie e i tool esterni).
- **Coding:** in questa fase gli sviluppatori eseguono lo sviluppo del codice di ciò che è stato richiesto nel documento di analisi funzionale.
- **Test:** i moduli di codice vengono prima testati singolarmente e poi, una volta integrati, vengono testati nel loro insieme.
- **Deploy:** in tale fase il software viene rilasciato al cliente
- **Manutenzione:** viene eseguita per migliorare, aggiornare e migliorare il prodotto finale.

I principali vantaggi derivanti dall'utilizzo di tale metodologia sono: l'utilizzo di un approccio sequenziale lineare, semplice da comprendere ed implementare, la possibilità di coordinare e gestire team di grandi dimensioni, infatti, la documentazione iniziale e le fasi di pianificazione consentono a gruppi numerosi di essere costantemente informati e muoversi verso un obiettivo comune, la possibilità di gestire i progetti in cui l'incertezza tecnologica non è elevata e non sono richiesti particolari cambiamenti nel corso dello svolgimento delle attività.

I principali svantaggi della metodologia sono:

- I requisiti devono essere perfettamente definiti fin dall'inizio del progetto;
- Il cliente ha possibilità di vedere il prodotto finito solo nelle fasi conclusive del progetto;
- È poco flessibile;
- Il tempo che intercorre tra l'avvio del progetto e il lancio sul mercato può essere elevato.

### **1.2.2 La metodologia Agile**

Tale metodologia, si contrappone al modello a cascata proponendo un approccio meno strutturato e focalizzato sull'obiettivo di consegnare al cliente, riducendo i tempi e con maggior frequenza, aiutando il team a rispondere all'imprevedibilità della costruzione del software. Utilizza sequenze di lavoro incrementali e iterative, comunemente note come sprint. Uno sprint è un periodo di tempo assegnato per una particolare fase di un progetto. La gran parte cerca di ridurre il rischio di fallimento sviluppando il software o le sue evolutive in intervalli di tempo limitati che prendono il

nome di iterazioni, la cui durata, generalmente, non supera le due o tre settimane. Ogni iterazione è un piccolo progetto a sé stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo incremento nelle funzionalità del software: pianificazione, analisi dei requisiti, progettazione, implementazione, test, deploy e manutenzione. Tra i vantaggi dell'utilizzo di Agile si possono riscontrare la facile adattabilità, ambiente rapido, rapporto stretto con il cliente, cultura del team. Se però Agile non è ben compreso né dal team, né dall'utente, potrebbe rivelarsi un'arma a doppio taglio.

### 1.2.3 Confronto tra la metodologia Waterfall e Agile

La *Tabella 1* mostra le differenze tra le principali variabili di interesse delle due metodologie.

Variabile	Waterfall	Agile
<b>Definizione dei requisiti</b>	Pianificazione meticolosa e rigidi	Possibili modifiche in itinere
<b>Controllo delle attività</b>	Incentrato sul processo	Incentrato sulle persone
<b>Ambito di utilizzo</b>	Grado d'incertezza basso	Grado d'incertezza alto
<b>Assegnazione dei ruoli</b>	Individuale	Team auto-organizzati
<b>Tipologia di comunicazione</b>	Formale	Informale
<b>Ruolo del cliente</b>	Importante	Critico
<b>Tempi di consegna</b>	Da 6 a 12 mesi	Ogni 2/4 settimane
<b>Individuazione delle problematiche</b>	Si scoprono alla fine	Individuati sempre
<b>Costi nel ciclo di vita</b>	Prima elevati, poi decrescenti	Sempre sostenuti

*Tabella 1 - Confronto Waterfall e Agile*

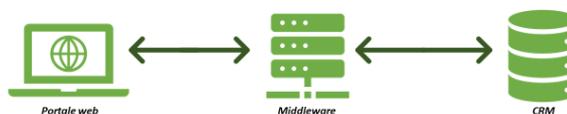
La decisione riguardo l'utilizzo di una metodologia rispetto ad un'altra deriva principalmente dalla maturità del prodotto finito e dalla comprensione dei risultati e dei requisiti del progetto nella fase di avvio del progetto, oltre che dai sistemi in gioco. La necessità di un approccio strutturato, di una pianificazione a priori e di produrre una numerosa documentazione a dimostrazione del lavoro svolto, sposa bene le esigenze dell'azienda e del cliente. Per tale ragione, la metodologia utilizzata all'interno del progetto seguito in Accenture è la metodologia Waterfall.

## 2 Il processo di realizzazione delle evolutive

### 2.1 I sistemi impattati

Il processo di realizzazione delle evolutive per il software in essere dell'azienda cliente può coinvolgere sia i sistemi di back end che i sistemi di front end proprietari o di fornitori terzi. Il front end rappresenta, in generale, la parte visibile di un software, quella con cui l'utente può interagire

mentre il back end è il componente che consente l'effettivo svolgimento di tali interazioni. I sistemi coinvolti nel progetto sono: Portale web, Middleware e CRM, come mostrato in *Figura 1*.



*Figura 1 - I sistemi impattati*

Il middleware è un intermediario tra i diversi elementi dell'infrastruttura, è software in grado di mettere in comunicazione i differenti strati architetturali e facilitare il compito di chi ha il ruolo di progettare i singoli componenti (ad esempio una pagina web). L'utilizzo del middleware, può consentire un più elevato livello di servizio per gli utenti, ed un più elevato livello di astrazione per i programmatori, può inoltre facilitare la manutenzione, la stesura e l'integrazione di applicazioni. Il CRM, invece, è una strategia aziendale che pone il cliente al centro del proprio lavoro, accelera il trasferimento di informazioni dal cliente all'organizzazione e viceversa in tutti i settori. Le applicazioni CRM servono a tenersi in contatto con la clientela, a inserire le loro informazioni all'interno del DB e a fornire loro modalità per interagire in modo che tali interazioni possano essere registrate e analizzate.

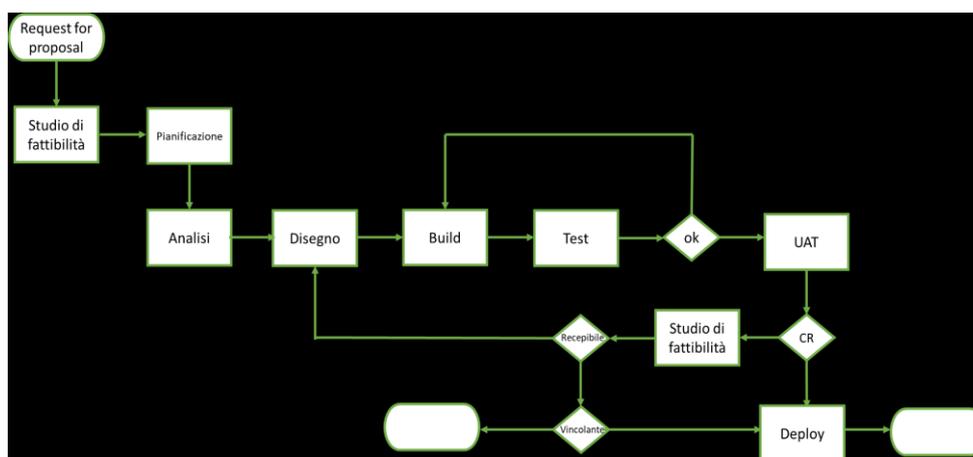
## **2.2 Le macro-fasi del processo di realizzazione delle evolutive**

Il processo di realizzazione delle evolutive è un processo molto complesso e per tanto viene rigidamente strutturato, in cui per ogni fase viene generalmente prodotto un output denominato deliverable. Esistono tre tipi di deliverable: il software prodotto, il servizio erogato e l'eventuale documentazione di supporto. L'evento trigger che avvia l'intero processo è costituito da una RFP (Request for propose), sotto forma di un documento denominato *Documento di requisito*, redatto dall'azienda cliente, mediante il quale viene richiesta l'implementazione di una nuova funzionalità o la modifica di una esistente. Il documento contiene il dettaglio del requisito che il fornitore dovrà implementare. In generale, i requisiti rappresentano le soluzioni ingegneristiche e tecniche per soddisfare un bisogno dei clienti. In seguito, l'azienda fornitrice provvede ad effettuare una valutazione della fattibilità dei requisiti raccolti. L'analisi è condotta da due punti di vista:

- Tecnico: si indaga la possibilità di sviluppare tecnicamente il requisito
- Economico/finanziaria: si indaga la convenienza economica della realizzazione del requisito e viene fatta una stima dell'ammontare di capitale necessario.

Durante tale fase è possibile che siano svolti degli incontri tra gli attori in gioco al fine di determinare il livello di comprensione del requisito. Al termine dello studio di fattibilità, viene stilato un

documento denominato *Schede tecnica*, grazie al quale il cliente valuta se la proposta di soluzione copre i desiderata, i costi rientrano nel budget e il piano proposto è in linea con i tempi attesi. Solo dopo queste attività preliminari il progetto viene avviato e si procede con l'ingaggio del team e con la pianificazione di dettaglio delle attività. Nella successiva fase, denominata analisi di dettaglio, sono previsti workshop con il cliente al fine di definire, nella maniera più dettagliata possibile, le nuove funzionalità da implementare per favorire l'avvio degli sviluppi. Per determinare come i requisiti verranno implementati sui sistemi e quale sarà la user experience, vengono redatte le specifiche funzionali nella fase di disegno. Tali specifiche rappresentano l'input della fase successiva, infatti, gli sviluppatori provvedono ad effettuare le dovute implementazioni sui sistemi coinvolti in base a come descritto all'interno delle specifiche funzionali. terminate le attività di sviluppo, i nuovi requisiti vengono rilasciati in ambiente di test per poter essere testati. Esistono diverse tipologie di test, nell'ordine: i System Test, test eseguiti in maniera standalone su ciascuna applicazione coinvolta nello sviluppo, gli Integration Test, test eseguiti end-to-end su tutte le componenti applicative coinvolte nel flusso di business, le quali possono essere gestite dal cliente o dal fornitore e User Acceptance test, test eseguiti con gli utenti al fine di verificare con gli stessi il corretto funzionamento delle implementazioni richieste. Durante lo svolgimento delle attività di UAT, qualora il comportamento dei requisiti oggetto di evolutive non sia coerente con quanto atteso dagli utenti, questi hanno la possibilità di richiedere delle CR (change request), le quali dovranno essere valutate, approvate e implementate dal team di sviluppo. Solo dopo aver raccolto l'approvazione da parte degli utenti, si procede con le attività di rilascio, dove i requisiti vengono trasferiti in ambiente di produzione. Il suddetto processo è mostrato in *Figura 2*.



*Figura 2 - Le macro-fasi del processo*

### 2.2.1 Studio di fattibilità

Le attività di tale fase iniziano con la ricezione di una richiesta di un documento di requisito proveniente dall'azienda cliente, in cui sono descritte diverse variabili: descrizione dell'esigenza, analisi as-is, esigenza da risolvere, impatti, vincoli, benefici attesi, linee guida di soluzione. Dopo aver valutato il documento, l'azienda fornitrice provvede ad effettuare lo studio di fattibilità del requisito

stilando un documento che contiene la proposta di soluzione e gli impatti sui sistemi. Al fine di redigere in maniera dettagliata il documento di fattibilità e definire correttamente l'effort necessario per l'implementazione del requisito, vengono svolte riunioni periodiche interne tra i membri responsabili dei sistemi impattati. Quando sono previsti impatti su sistemi esterni è necessario coinvolgere anche i fornitori terzi di tali sistemi. L'effort totale sarà dunque dato dalla somma degli sforzi necessari ad implementare il requisito sui vari sistemi, calcolato in giornate/uomo. Al fine di evitare ritardi, la stima, deve tener conto di eventuali problemi di contingency.

## 2.2.2 Pianificazione

Lo scopo della pianificazione è quello di strutturare le attività di progetto rispettando le tempistiche condivise ed i vincoli presenti, come la disponibilità di risorse, infrastrutture, ecc.

Tale fase include la definizione dello scope, la definizione del budget, vincoli e le risorse, la durata delle attività, i ruoli e le responsabili, la conferma del Blueprint della soluzione (possono essere presenti più alternative). Il business Blueprint è un documento in cui viene descritto il processo, la cui stesura viene solitamente fatta dall'azienda fornitrice che, interfacciandosi con il cliente, identifica le esigenze funzionali dettagliate e propone la soluzione da sviluppare. L'approvazione è una milestone di progetto, definendo l'avvio del progetto. In generale, il tempo che solitamente intercorre tra l'attività di pianificazione e quello di rilascio può superare i sei mesi. Durante le attività di pianificazione, il project manager è il principale attore coinvolto in quanto è il responsabile delle relazioni con i clienti. I requisiti per cui è stato valutato positivamente lo studio di fattibilità, vengono accorpati all'interno di un piano detto *Piano di release* per gestire in maniera efficiente, in termini di costo, tempo e qualità, tutto il processo aziendale (dell'azienda cliente) di change management. Con il termine release, viene indicato il rilascio di una nuova specifica funziona di un software, resa disponibile agli utenti finali. Una release può essere classificata in base alla sua dimensione e si distinguono major release (riguardano evoluzioni sostanziali rispetto alla precedente e inglobano al loro interno più requisiti) e minor release (correzione di bug). Il processo di realizzazione di una major release è un processo molto complesso e per tanto viene rigidamente strutturato. Lo scheduling delle attività del piano di release può essere descritto mediante l'utilizzo di un diagramma di Gaant, come rappresentato in *Figura 3*:



*Figura 3 - Esempio di pianificazione - Diagramma di Gaant*

Il piano di progetto prodotto al momento di kick-off è generalmente di alto-livello in quanto viene arricchito con i dettagli con il susseguirsi delle fasi. Infatti, le milestone potrebbero subire variazioni rispetto a quanto pianificato a causa di ritardi o imprevisti.

### 2.2.3 Analisi di dettaglio

Le attività in tale fase sono: la valutazione, in cui si revisionano i requisiti organizzando meeting con gli utenti in modo da comprendere al meglio lo stato attuale e i requisiti futuri, gli impatti dei requisiti sul sistema e le differenze rispetto al modello attuale, la progettazione del nuovo work flow in cui si definisce il processo to-be, ovvero come la soluzione verrà proposta, tenendo in considerazione l'impatto sui sistemi attuali e la revisione dove si deve confermare che i requisiti siano coerenti e integrabili all'interno del sistema, tutti i punti aperti siano adeguatamente affrontati e discussi. Tali attività vengono eseguite mediante lo svolgimento di workshop tra l'azienda cliente e il fornitore. Tale fase può durare anche diversi mesi. Possono comunque emergere diverse criticità durante tale fase, infatti, per esempio, si può non comprendere quali siano le esigenze del cliente, oppure può succedere le esigenze del cliente non sono in linea con la fattibilità del requisito, o ancora per far sì che il requisito sia implementabile.

### 2.2.4 Disegno

Nella fase di disegno, deve essere trasferita all'interno delle specifiche funzionali la presentazione della soluzione arricchita con i dettagli dell'utente. Durante tale fase è necessario assicurarsi che la progettazione non superi i requisiti funzionali. Viene redatto il documento *Specifiche funzionali*, il quale ha il compito di descrivere come le implementazioni funzionano all'interno dei sistemi. In altre parole, descrivono ciò che farà il software e i suoi moduli grazie alle future implementazioni dei nuovi requisiti. Dunque, per tale fase i deliverable in input sono le specifiche funzionali in essere mentre, in output quelle aggiornate contenenti le informazioni sui nuovi requisiti. La *Figura 4* mostra il dettaglio dei deliverable.



**Figura 4 - Deliverable I/O**

Le attività svolte all'interno di tale fase sono mostrate nella

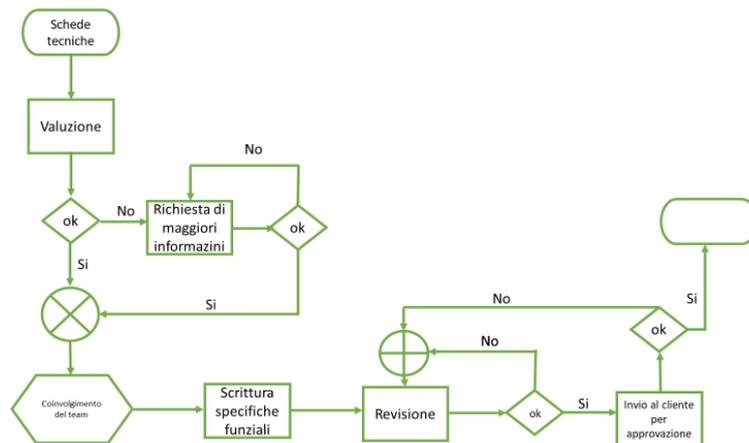


Figura 5 - Attività funzionali nella fase di Design

### 2.2.5 Build

Nelle fase di build vengono eseguite le attività di sviluppo del codice da parte degli sviluppatori sulla base di quanto appreso all'interno delle specifiche funzionali e l'esecuzione dei system test. Dall'altro lato, il team funzionale, in tale fase è impegnato nelle attività di stesura del testbook, infatti, per ogni requisito, vengono previsti diversi scenari di test al fine di verificarne la corretta funzionalità. Dopo avere definito tutti i possibili scenari di test, dal team funzionale viene eseguita l'attività di pianificazione dei test secondo i seguenti step: calcolo del numero di test totali, divisione del numero di test totali sulle giornate uomo disponibili per le attività, assegnazione degli scenari di test al team. Il carico di lavoro dello svolgimento dei test deve essere distribuito in modo tale da risultare maggiore all'inizio della fase di test in modo da avere la possibilità di effettuare nuovi cicli di test a causa della presenza di anomalie riscontrate durante l'esecuzione. Infine, prima dell'inizio delle attività di testing, il testbook deve essere inviato al cliente per poter essere approvato. Il dettaglio delle attività è mostrato nella *Figura 6*.

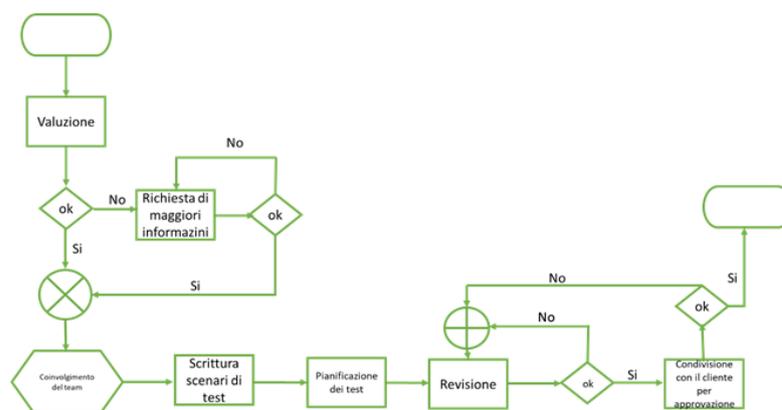


Figura 6 - Attività funzionali fase di Build

### 2.2.6 Test

In tale fase vengono svolti gli Integration Test. L'esecuzione dei test spetta al team funzionale. Nella casistica in cui siano presenti problematiche (bug) per i requisiti, queste vengono tracciate mediante l'utilizzo di una piattaforma di *Bug Tracker* in modo da mantenere tracciabilità degli errori riscontrati.

Inoltre, tale applicativo permette di fornire i dettagli dei bug individuati e consente agli sviluppatori di replicarli e di poterli risolvere. Solo dopo aver riverificato il corretto funzionamento del sistema, l'esito del test sarà considerato positivo.

### 2.2.7 UAT

Terminata la fase di test, l'attività successiva è quella degli UAT (User Acceptance Test), ovvero i test che vengono svolti insieme agli utenti per verificare il corretto funzionamento dei nuovi requisiti all'interno del sistema. Tale fase prevede solo un numero limitato di test, precedentemente definito. Per tale ragione è opportuno svolgere i test che mettono in evidenza le funzionalità primarie dei requisiti. È possibile che, durante lo svolgimento degli UAT, gli utenti possano richiedere una CR (change request) qualora il comportamento non sia in linea con quanto atteso. Le CR vengono valutate dagli sviluppatori solo una volta terminata l'attività di UAT. Qualora la CR fosse recepitibile, verranno apportate le opportune modifiche ai sistemi coinvolti e successivamente testate per verificarne il corretto funzionamento. Qualora, invece, la CR non fosse recepitibile o per motivi tecnici o a causa delle tempistiche stringenti per poter rieseguire tutte le attività, questa verrà censita all'interno di un documento per mantenerne la tracciabilità. Il dettaglio delle attività è mostrato nella

Figura 7.

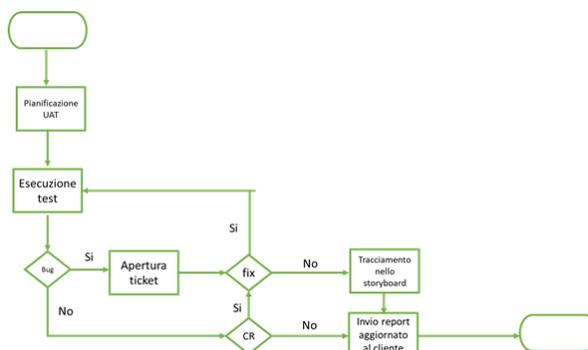


Figura 7 - Attività fase UAT

### 2.2.8 Deploy

Prima di procedere con le attività di deployment, è necessario che venga preparato un piano detto *Piano di Rolloout* che ha il compito di scandire le attività durante il rilascio. Tale piano deve essere approvato dal cliente. Ricevuta l'approvazione, viene effettuato, in ambiente di produzione, il deploy implementazioni sviluppate. L'attività, solitamente, si svolge in una finestra temporale al di fuori del normale orario lavorativo, in modo da avere il minor impatto possibile sui sistemi. Una volta effettuato il deploy, un membro del team funzionale esegue i sanitycheck, ovvero dei test volti a verificare che le funzionalità principali dei requisiti in ambiente di produzione siano state correttamente implementate. Qualora i sanitycheck avessero esito negativo, allora si dovrà procedere con il rollback, gli sviluppatori dovranno ripristinare in ambiente di produzione lo stato precedente e dovrà essere riprogrammata una nuova data di rilascio. Ciò comporterà, una volta individuato e risolto il problema, una nuova sessione di test al fine di verificare la corretta esecuzione

delle operazioni e la pianificazione di una nuova data per poter effettuare le implementazioni richieste. Indipendente dall'esito, tutte le parti interessate dovranno essere tempestivamente informate sull'esito della fase. I deliverable sono mostrati nella *Figura 8*.



*Figura 8 - Deliverable fase di Deploy*

Terminate le attività di deploy, le successive attività di manutenzione vengono assegnate al team di application maintenace.

### 3 Conclusioni

Il processo di realizzazione delle evolutive è terminato con il rilascio in produzione dei requisiti oggetto della release con un Go-Live posticipato rispetto a quanto pianificato. Anche se con ritardo, il desiderato dell'utente e di tutti gli attori coinvolti nelle diverse attività è risultato soddisfatto. Oltre a semplici ritardi nell'esecuzione delle attività, o al manifestarsi di CR durante la fase di UAT, la data di Go-Live della release è stata soggetta a variazione a causa di numerosi fattori. Per esempio, è stato necessario coinvolgere più fornitori esterni, i quali hanno aumentato il livello di complessità del processo. Tali fornitori, infatti, hanno avuto ritardi nello svolgimento delle attività riguardanti più requisiti. Per alcuni requisiti, è stata posticipata la data di Go-Live creando delle code della release.

Per un requisito, sono state evidenziate delle problematiche che uno dei fornitori non è stato in grado di risolvere. Per tale casistica è stato adoperato un workaround, ovvero, l'utilizzo di un metodo temporaneo per raggiungere una soluzione al fine di portare il requisito in produzione. Inoltre, per una particolare implementazione, a causa di esigenze esterne, quali il recepimento di particolari direttive o leggi, al fine di evitare il rischio di incorre in sanzione, è stato necessario rilasciarla in anticipo rispetto all'interna release. Il processo descritto, è un processo che si ripete ciclicamente, infatti, ad oggi sono al vaglio dello studio di fattibilità, nuovi requisiti per evolutive future. Grazie alla possibilità di prendere parte ad un progetto nell'ambito IT, è stato possibile conoscere l'interno ciclo di vita di un software e il processo di realizzazione degli interventi di evolutive. Inoltre, l'approdo in un team di professionisti operanti nel settore da tempo ha accelerato il processo di crescita sia dal punto di vista delle competenze tecniche che tacite. Infine, l'ingresso diretto nel mondo del lavoro ha permesso di sfruttare il background accademico di questi anni e di integrarlo con nuove conoscenze e competenze acquisite nel corso del periodo di formazione che saranno fondamentali in futuro.