



*Istituto Nazionale di Previdenza per i Dipendenti  
dell'Amministrazione Pubblica*

## CAPITOLATO TECNICO

PER UN SERVIZIO DI MANUTENZIONE ED EVOLUZIONE  
DELLA COMPONENTE ISTITUZIONALE DEL SISTEMA  
INFORMATIVO INPDAP.

### ALLEGATO 1

“Linee guida al Ciclo di Vita del Software”

Dicembre 2009

## SOMMARIO

<b>Premessa .....</b>	<b>5</b>
<b>1. Le fasi e i processi del ciclo di vita del software. ....</b>	<b>10</b>
<b>2 Processi primari del ciclo di vita del software .....</b>	<b>11</b>
<b>2.1 Fase di Ingegneria dei requisiti .....</b>	<b>11</b>
Obiettivi del Processo .....	12
Risultati attesi .....	12
Uscita dalla Fase .....	12
Attori/Ruoli Aziendali coinvolti .....	12
Work products – input.....	12
Output .....	12
Attributi dell'oggetto Requisito.....	13
Modelli usati nella descrizione del Requisito.....	14
Attività .....	15
Attività .....	15
Documento di Definizione dei Requisiti.....	17
<b>2.3 Fase di Disegno Tecnico .....</b>	<b>19</b>
Obiettivi del Processo.....	19
Risultati attesi .....	19
Uscita dalla Fase .....	20
Attori/Ruoli Aziendali coinvolti .....	20
Work products – input.....	20
Output .....	20
Attributi dell'Oggetto Disegno (Specifiche Tecniche del Sistema o Disegno del Componente) .....	21
Modelli usati nelle Specifiche Tecniche.....	21
Attività .....	22
<b>2.4 Fase di Costruzione.....</b>	<b>26</b>
Obiettivi del Processo.....	26
Risultati attesi .....	26
Uscita dalla Fase .....	26
Attori/Ruoli Aziendali coinvolti .....	26
Work products – input.....	26
Output .....	27
Attributi dell'oggetto Componente Applicativo.....	27
Modelli usati nella fase .....	28
Attività .....	28
<b>2.5 Avviamento in Esercizio .....</b>	<b>31</b>
Obiettivi del Processo.....	31
Risultati attesi .....	31
Uscita dalla Fase .....	31
Attori/Ruoli Aziendali coinvolti .....	31
Work products – input.....	32
Output .....	32

Attributi dell'oggetto Sistema .....	32
Modelli usati per ambiente di deployment .....	33
Attività .....	34
<b>2.6 Manutenzione .....</b>	<b>38</b>
Obiettivi del Processo .....	38
Risultati attesi .....	38
Uscita dalla Fase .....	38
Attori/Ruoli Aziendali coinvolti .....	38
Work products – input.....	38
Output .....	39
Attributi dell'oggetto Difetto/Fix.....	39
Attività .....	40
<b>3 Modelli .....</b>	<b>42</b>
<b>3.1 Use Case.....</b>	<b>42</b>
<b>3.2 Sequence Diagram .....</b>	<b>43</b>
<b>3.3 Diagramma Entity-Relationship o di Classi di Analisi .....</b>	<b>44</b>
<b>3.4 Modello dei Componenti .....</b>	<b>45</b>
<b>3.5 Activity Diagram .....</b>	<b>46</b>
<b>3.6 Diagramma di Deployment .....</b>	<b>47</b>
<b>4 Processi Trasversali e tools di supporto .....</b>	<b>49</b>
<b>4.1 Processo di gestione della Configurazione .....</b>	<b>50</b>
Obiettivi del Processo.....	50
Risultati attesi .....	50
Attori/Ruoli Aziendali coinvolti .....	50
Work products – input.....	50
Output .....	51
Attributi dell'oggetto di configurazione (Configuration Item).....	51
Modelli usati nel processo .....	51
Attività .....	52
<b>4.2 Ingegneria dei Test .....</b>	<b>54</b>
Obiettivi del Processo.....	56
Attori/Ruoli Aziendali coinvolti .....	56
Risultati attesi .....	56
Work products – input.....	56
Output (modelli) .....	57
Attributi dell'oggetto Test .....	57
Modelli usati nella fase .....	58
Attività .....	59
<b>4.3 Gestione dell'Architettura Applicativa e Tecnica .....</b>	<b>61</b>
Obiettivi del Processo.....	61
Risultati attesi .....	61

Attori/Ruoli Aziendali coinvolti .....	61
Work products – input.....	61
Output .....	61
Attributi dell'oggetto di Architettura.....	61
Modelli usati nel processo .....	62
Attività .....	63
<b>4.4 Processo di Gestione della Documentazione .....</b>	<b>64</b>
Obiettivi del Processo.....	64
Risultati attesi .....	64
Uscita dalla Fase .....	64
Attori/Ruoli Aziendali coinvolti .....	64
Work products – input.....	65
Output .....	65
Attributi dell'oggetto "Documento" .....	65
Attività .....	66
<b>4.5 Processo di Riuso (Gestione Asset Riusabili).....</b>	<b>68</b>
Obiettivi del Processo.....	68
Risultati attesi .....	68
Uscita dalla Fase .....	68
Attori/Ruoli Aziendali coinvolti .....	68
Work products – input.....	68
Output .....	69
Attributi dell'oggetto "Asset".....	69
Attività .....	69

## Premessa

Questo documento contiene le linee guida alla gestione del ciclo di vita del software (CVS) in INPDAP, per quanto riguarda i prodotti sviluppati ad hoc. Il documento non è esaustivo nel trattare tutte le attività previste in un CVS e ulteriori raccomandazioni potranno essere definite in seguito da INPDAP. Inoltre, le prescrizioni contenute in questo documento vanno in certi casi adattate al contesto di utilizzo, secondo modalità che devono essere di volta in volta definite. Preliminarmente, vengono forniti dei richiami metodologici che aiutano a definire il vocabolario che verrà utilizzato nel testo e alcuni concetti base che saranno poi sviluppati nelle linee guida.

La produzione del software avviene seguendo un ciclo produttivo non dissimile da quello usato per gli altri prodotti industriali: un committente definisce delle esigenze (dei “**requisiti**”), un progettista disegna una soluzione tecnica che soddisfa queste esigenze, sulla base di tale progetto una “fabbrica” costruisce un prodotto che, prima di essere consegnato al committente, viene “verificato” (sottoposto a dei “test”) per accertare che quanto realizzato corrisponda al progetto tecnico e soddisfi le esigenze del committente. Il prodotto realizzato viene quindi consegnato al cliente (che può effettuare a sua volta un “collaudo” per verificare se quanto consegnato risponde a quanto richiesto) e viene “messo in esercizio” (il software è reso disponibile agli utenti per l'utilizzo, nell'ambiente di lavoro di questi utenti). Dopo la consegna, il prodotto viene sottoposto a una “manutenzione” per correggere gli eventuali “difetti” che manifesta nell'uso, oppure per migliorarne le prestazioni o per permettergli di funzionare anche quando dovesse cambiare l'ambiente tecnologico per lavorare nel quale era stato progettato. Il prodotto può essere anche modificato dopo la consegna al committente per metterlo in grado di svolgere ulteriori funzioni per le quali non era stato progettato (quindi per rispondere a “nuove esigenze” del committente). In questo caso si parla di “manutenzione evolutiva” del prodotto.

Durante il ciclo lavorativo sopra sinteticamente schematizzato (definizione esigenze – progettazione soluzione tecnica – costruzione prodotto – test – consegna e messa in esercizio – manutenzione), il prodotto software attraversa diversi **stadi** di lavorazione: all'inizio è solo un documento che descrive delle esigenze e solo via via diventa del codice sorgente. Ognuna delle fasi lavorative dell'iter produttivo del software produce dei **semilavorati** (deliverables) specifici. Tra questi semilavorati (e quindi tra le diverse fasi lavorative) esistono dei precisi legami di dipendenza (di tipo “input-output”): e.g. un disegno tecnico di un componente software non può essere definito se prima non si è definito l'insieme di requisiti che questo componente deve soddisfare. Molti dei semilavorati tipici del ciclo di produzione del software un sono composti da vero e proprio software ma sono dei documenti, che rappresentano dei **modelli** del funzionamento del sistema software da realizzare: si può anzi affermare che la costruzione del prodotto software procede attraverso la definizione di modelli sempre più di dettaglio del funzionamento del sistema da realizzare, fino a che i modelli sono sufficientemente precisi da consentire di realizzare un oggetto software funzionante a partire da essi.

Le **fasi** di lavoro necessarie a produrre un software sono di norma piuttosto complesse in quanto composte di diverse **attività** (a loro volta fatte di vari “**task**” o compiti più elementari), che si susseguono anch'esse sulla base di vari legami di dipendenza. Ogni fase ha proprie

attività specifiche, che operano sui semilavorati tipici della fase. Ogni attività ha un ruolo preciso nella trasformazione dei semilavorati propri della fase. Un insieme correlato di attività, svolte in modo integrato al fine di raggiungere un determinato obiettivi trasformando un dato input in un dato output può essere chiamato “**processo**”.

La complessità intrinseca di un prodotto software non consente però, nella realtà dei progetti, di procedere alla costruzione del software eseguendo secondo una rigida sequenza temporale le fasi di lavoro sopra individuate. Normalmente, la produzione del software avviene infatti in maniera “**iterativa e incrementale**” nel senso che:

- si ammette la possibilità che le esigenze (requisiti) del committente possano modificarsi durante il ciclo di produzione stesso (il committente di norma all'inizio del progetto non ha ancora consapevolezza di tutte le potenzialità del software e scopre parte delle sue esigenze solo vedendo le soluzioni tecniche che gli vengono proposte). Perciò, si procede con la costruzione del software in maniera incrementale, partendo da alcuni componenti o prototipi, che poi vengono fatti evolvere fino a che il committente non è soddisfatto del “modello” del sistema che gli viene presentato; in questo modo si minimizza l'impatto (in termini di costo) delle modifiche da apportare al progetto a seguito dei ripensamenti del committente;
- la metodica più diffusa prevede di scomporre il sistema software da realizzare in componenti più piccoli, e di eseguire la sequenza delle varie fasi di lavorazione in modo indipendente per ogni componente (per quanto possibile), sfruttando i possibili parallelismi e “localizzando” le eventuali modifiche apportate in corso d'opera alla progettazione;

La produzione del software avviene pertanto seguendo una successione ordinata di processi/attività, correlati tra loro da legami di dipendenza, sequenza applicata però non all'intero sistema software da realizzare, ma ai singoli componenti nei quali il sistema può essere scomposto. Questo approccio (che si può chiamare “metodologia”) richiede, perciò, come primo passo di procedere alla definizione della “WBS” (Work Breakdown Structure) del sistema da realizzare, ovvero alla sua scomposizione in un certo numero di componenti più di dettaglio, rappresentabile in un diagramma gerarchico; non c'è una regola fissa che guida la “granularità” della scomposizione, l'importante è che ogni componente individuato nella WBS sia chiaramente identificabile quanto a caratteristiche, impegno necessario allo sviluppo, legami con gli altri componenti.

In un “cantiere” per lo sviluppo del software è necessario svolgere anche altre attività, oltre quelle fin qui individuate (che potremmo chiamare “**primarie**”). Come detto, ogni attività produce dei documenti e dei semilavorati, dei quali vanno gestiti l'archiviazione, le modifiche, le versioni etc.. E' necessario quindi prevedere una “gestione della documentazione” così come una “gestione della configurazione” di quanti via via prodotto nel ciclo di lavorazione. Si è poi accennato alla possibile variabilità dei requisiti dati dal committente: per evitare problemi di fraintendimenti e di corretto passaggio di informazioni tra le fasi lavorative è necessario anche prevedere una “gestione dei requisiti” (*change request management*), che ne assicuri la archiviazione, tracciabilità, integrità, coerenza etc. I requisiti sono strettamente correlati ai test

da effettuare sui semilavorati, che devono verificare se in quanto viene costruito sono rispettate le richieste del committente: va prevista perciò in un cantiere del software anche la “gestione dei casi di test”, che devono essere sempre allineati alle configurazioni degli oggetti da testare e dei requisiti da soddisfare.

La gestione dei test è una attività fondamentale per una corretta esecuzione del ciclo di vita del software. Le attività di gestione dei test sono infatti articolate: pianificazione e controllo dei test, scelta delle condizioni di test, progettazione dei test case e script, controllo dei risultati, valutazione dei criteri di uscita, reportistica sul processo di testing e sul sistema sotto test, e la finalizzazione o chiusura (ad esempio, dopo che una fase di test è stata completata). Il testing include poi anche la revisione dei documenti (compreso il codice sorgente) ed una analisi statica.

Il test può poi avere differenti finalità:

- trovare difetti;
- acquisire confidenza circa il livello di qualità ed ottenere opportune informazioni;
- prevenire i difetti.

In particolare, dedicarsi alla progettazione dei test nelle fasi iniziali del ciclo di vita (attraverso una progettazione di test di alto livello basata sui requisiti) può aiutare a prevenire difetti prima che essi vengano introdotti nel codice.

Tutte le attività sopra individuate gestiscono in effetti il ciclo di vita di una tipologia di semilavorato del processo produttivo. Questi semilavorati vengono trasformati nel tempo dalle attività primarie. Perciò, queste attività di supporto alla gestione dei semilavorati sono dette “**trasversali**” a quelle primarie, nel senso che non sono legate strettamente a una sola di esse o a una fase specifica di lavorazione del software, ma supportano il corretto svolgimento dell'intero progetto. L'impegno da prevedere nelle attività trasversali non è costante, ma varia da fase a fase, in funzione della consistenza dei semilavorati che gestiscono.

Infine, in un cantiere di sviluppo software esistono anche delle attività di natura “**organizzativa**” parimenti indispensabili al buon esito del progetto: le **risorse** (professionali e tecnologiche) che operano nel progetto vanno infatti pianificate, il loro lavoro va coordinato, organizzato e controllato, vanno tenuti sotto controllo i costi sostenuti (rispetto al budget disponibile) e i rischi di non riuscire a raggiungere gli obiettivi attesi. Queste sono attività proprie del **Project Management**, la funzione alla quale è delegata la gestione operativa delle risorse che operano sul progetto. Inoltre, come dimostrato dalla letteratura sull'ingegneria del software, le probabilità che un progetto vada a buon fine sono massimizzate se il lavoro è svolto seguendo “regole” e procedure definite (delle metodologie), se vengono effettuate verifiche sulla corretta applicazione di queste regole e procedure, se vengono rilevate misure di efficienza ed efficacia dei processi lavorativi e tali misure sono utilizzate per migliorare il modo di lavorare. Vanno perciò previste in un progetto anche delle attività di Audit, Verifica, Review sul corretto svolgimento delle procedure lavorative, sia al fine di rilevare casi di applicazione non conforme delle regole, sia per trovare azioni correttive, sia per individuare possibili spazi di miglioramento.

Tutte le attività di un ciclo di produzione del software, sia primarie, sia trasversali di supporto che organizzative, per la loro intrinseca complessità e per la quantità di legami di dipendenza che hanno tra di loro, devono essere svolte, non appena le dimensioni del progetto superano una certa soglia dimensionale, utilizzando dei **tools** specifici.

In definitiva, il ciclo di produzione del software può essere schematizzato come un framework, un “modello” di lavoro (o un “metodo”), caratterizzato da una sua sintassi (gli elementi che lo compongono), dalla sua semantica (il significato attribuito a questi elementi), nonché dalle relazioni che legano questi elementi tra loro (il “ruolo” che svolgono nel framework).

In letteratura, non esiste una rappresentazione univoca di questo framework. Sebbene i componenti di base del procedimento con il quale si produce il software siano quelli sopra descritti, esistono diverse tassonomie e classificazioni per le attività da svolgere, i semilavorati da produrre, i modelli con i quali rappresentare i diversi aspetti di un sistema software nei suoi vari stadi di lavorazione. I riferimenti più diffusi sono lo standard **ISO/IEC 12207 (2008)**, che definisce i processi del ciclo di vita del software (43 diversi organizzati in 7 gruppi), soprattutto in termini di “cosa” va fatto per produrre il software (definendo anche una terminologia di riferimento per attività, semilavorati, modelli), lo **SWEBOK** (Software Engineering Book of Knowledge), nato da un progetto IEEE ed ora diventato standard **ISO/IEC 19759**, che definisce 11 macro processi, il **CMMI** (Capability Maturity Model Integrated) frutto di un progetto del Software Engineering Institute della Carnegie Mellon University di Pittsburgh, USA, che definisce 22 processi organizzati in 4 categorie, lo standard **IEEE 1074**, che definisce 16 processi, organizzati in 4 classi, **ISBSG** che definisce 20 processi organizzati in 4 gruppi. Altri riferimenti meno diffusi sono lo standard ESA-PSS-05-0 e lo standard DOD 2167A. Un possibile riferimento è anche lo standard **ISO 90003**, che definisce come applicare le prescrizioni dello standard ISO 9001 al settore dello sviluppo del software. In questo senso, fornisce raccomandazioni su come svolgere circa 20 processi connessi alla realizzazione del prodotto (organizzati in 6 gruppi), 9 processi per la misura, analisi dei dati e uso dei dati rilevati al fine di migliorare, più alcuni processi per la gestione delle risorse di progetto e per il management.

Il metodo **UP** (Unified Process, ideato da I.Jacobson, G.Booch e J.Rambaugh, gli autori di UML) definisce 4 fasi di lavorazione del software, che si ripetono più volte in maniera sequenziale in un ciclo iterativo e incrementale di sviluppo, ognuna delle quali termina con una *milestone* (un insieme intermedio di artefatti - ad es. una specifica progettuale - che possono essere sottoposti a una verifica). In ogni fase sono previste 9 attività (o “discipline”), le stesse in tutte e quattro le fasi, ma svolte con impegno diverso da fase a fase e non necessariamente da eseguire in maniera sequenziale. 6 attività riguardano lo sviluppo vero e proprio del prodotto software e 3 sono di supporto, secondo uno schema concettuale non dissimile da quello ISO/IEC 12207. UP definisce quindi 9 modelli del sistema software che vanno realizzati nel ciclo di produzione, ognuno deputato a descrivere un aspetto specifico del software.

Il metodo “**a spirale**” definito da B.Boehm, anch'esso di tipo iterativo incrementale, prevede infine 4 (o 6) fasi di lavorazione che si susseguono e 9 attività da svolgere in queste fasi, in maniera non dissimile dalla sequenza UP.



Come detto, in un cantiere di sviluppo del software è fondamentale poter **verificare** se il framework metodologico adottato dal team di lavoro sia eseguito correttamente. Questo tipo di verifica non va tanto a valutare le caratteristiche dell'output delle varie attività svolte, quanto che tali attività siano definite in apposite procedure (in documenti) in maniera completa, comprensibile, ripetibile, così da assicurare con ragionevole certezza che la messa in pratica delle attività porterà a produrre output della qualità attesa (si afferma in questo caso che il processo ha la "capacità" potenziale di produrre l'output atteso). Si tratta quindi di effettuare delle verifiche di "processo", per lo svolgimento delle quali sono disponibili in letteratura diversi riferimenti, quali lo standard **ISO/IEC 15504**, che guida alla verifica della corretta definizione dei processi definiti nello standard ISO/IEC 12207, del metodo **SCAMPI**, che guida alla verifica della corretta definizione dei processi definiti nel **CMMI**, dello standard **ISO 19011**, che guida alla verifica della corretta applicazione delle prescrizioni contenute negli standard della serie ISO 9000.

Ciò premesso, qui di seguito si riportano le prescrizioni base e generali per lo svolgimento di alcuni dei principali processi e delle attività del ciclo di vita del software in INPDAP. Sono trattati sia i processi primari che quelli trasversali del ciclo di vita. Vengono anche fornite indicazioni sui modelli da utilizzare per la rappresentazione del sistema software nelle varie fasi di realizzazione.

Alla fine del documento sono quindi riepilogati i principali documenti che guidano con maggior dettaglio alla effettuazione delle attività del ciclo di vita, applicando le regole generali a contesti specifici. Questi documenti sono disponibili a richiesta presso la DCSI INPDAP.

## 1. Le fasi e i processi del ciclo di vita del software.

---

Le fasi primarie del ciclo di produzione del software sono le seguenti:

1. Ingegneria dei requisiti
2. Disegno tecnico del Software
3. Costruzione del software
4. Avviamento in Esercizio
5. Manutenzione

Queste fasi sono composte ognuna da diversi processi, attività e task, come specificato nel seguito.

A supporto di queste fasi primarie devono essere attuati i seguenti processi trasversali:

1. Gestione dell'Architettura Applicativa e Tecnica
2. Ingegneria del Test
3. Gestione della Documentazione
4. Gestione della Configurazione
5. Gestione della Qualità del Software
6. Gestione degli Asset e del Riutilizzo

e i seguenti processi organizzativi:

1. Gestione del progetto
2. Gestione della qualità del progetto

Per ognuno dei processi del ciclo di vita del software sono di seguito definiti questi elementi:

- a) lo scopo od obiettivo (purpose) per il quale si esegue il processo
- b) i risultati attesi (outcomes) a seguito dell'attuazione del processo
- c) il semilavorato (o i semilavorati, se più di uno) in input al processo (i modelli od anche work product)
- d) le trasformazioni da eseguire su questi semilavorati
- e) l'output (o gli outputs) da produrre (i modelli)
- f) le attività da svolgere con i relativi task
- g) le verifiche da prevedere per accertarsi che il processo sia stato eseguito correttamente

## 2 Processi primari del ciclo di vita del software

---

### 2.1 Fase di Ingegneria dei requisiti

Il requisito è l'elemento base per la gestione del progetto da parte del Responsabile di Area dell'Istituto.

Il processo raccoglie i bisogni degli Utenti e degli altri "stakeholders", ne definisce le priorità di realizzazione nel piano di rilascio e li fa evolvere fino al livello di specifiche approvabili dagli utenti.

I requisiti sono classificati in quattro tipologie:

1. Funzionali: i requisiti funzionali descrivono le necessità degli utenti finali in termini di supporto applicativo ai processi dell'Istituto. I requisiti sono rappresentati mediante Scenari (Casi d'Uso). Un particolare Caso d'Uso è quello che rappresenta il "sistema" ed i suoi "Attori", vale a dire il Diagramma di Contesto dell'applicazione (vedi allegato).
2. Vincoli Funzionali: sono i vincoli che dovranno essere rispettati nel soddisfare i requisiti funzionali e discendono da regole e/o politiche aziendali. Di seguito sono riportate le tipologie più comuni di Vincoli Funzionali:
  - a. Direttive Aziendali (ad es. orari di lavoro)
  - b. Fattori costitutivi nel dominio funzionale considerato (Entità logiche, o classi di analisi, del dominio applicativo - ad. es. ogni applicazione pensionistica dovrà trattare i concetti di beneficiario, prestazione, metodo di pagamento, ecc.-)
  - c. Normativi (leggi e regolamenti)
  - d. Integrazioni con i Sistemi Istituzionali e di Autogoverno (sistemi esistenti con cui si deve necessariamente scambiare informazioni)
3. Non funzionali: i requisiti non funzionali definiscono il come i requisiti funzionali dovranno essere soddisfatti in termini di 'prestazioni ed usabilità dell'applicazione. Le principali dimensioni dei requisiti funzionali sono le seguenti:
  - a. Capacità e Volumi (ad. es. Numero di utenti concorrenti)
  - b. Prestazioni (ad. es. Tempi di risposta in specifiche situazioni)
  - c. Disponibilità (disponibilità del sistema)
  - d. Usabilità (livello di training necessario per l'uso)
  - e. Accessibilità (es. rispetto della normativa Stanca)
  - f. Scalabilità (possibilità di evolvere con l'aumento del carico)
  - g. Affidabilità (capacità di continuare a funzionare, anche se in maniera degradata a fronte di certi errori)
  - h. Sicurezza fisica e prevenzioni infortuni (se si tratta di sistemi speciali)

4. Regole di sicurezza: questi requisiti determinano le condizioni in cui il funzionamento dell'applicazione espone l'istituto ad un livello di rischio accettabile per politica, requisiti interni e/o di legge.

### **Obiettivi del Processo**

- Stabilire e gestire un accordo con gli utenti e le altre figure coinvolte su cosa deve fare il sistema.
- Fornire al gruppo di sviluppo una migliore comprensione dei requisiti di sistema.
- Definire i confini del sistema (delimitarlo).
- Fornire una pianificazione dei rilasci.
- Fornire una valutazione dei costi e del tempo necessario per sviluppare il sistema.
- Definire ad alto livello le interfacce del sistema, mettendo a fuoco le necessità e gli obiettivi degli utenti.

### **Risultati attesi**

Definizione delle specifiche del sistema in costruzione.

### **Uscita dalla Fase**

1. Le Specifiche risultano approvate dagli Utenti.
2. I Costi e i piani di rilascio sono approvati dal Management dell'INPDAP

### **Attori/Ruoli Aziendali coinvolti**

Referente Applicativo: Responsabile DCSIT del Progetto o Area Applicativa

Utenti (stakeholders): Rappresentanti delle linee operative che utilizzeranno il sistema

Analista: Figura tecnica specialista del dominio applicativo

Architetto Applicativo: Figura tecnica specialista di sistemi software

### **Work products – input**

[PL] Principi, Linee Guida e Standards di riferimento

### **Output**

[RQ] Requisiti

Output indiretti del processo sono:

[UC] Use Case Model

[SQ] Sequence Diagram

[AD] Activity Diagram/Process Diagram

[IU] Specifiche interfacce utente

[ER] Modello concettuale dei dati

[PR] WBS e Gantt del progetto

[TC] Test Cases (Tabella dei Test)

### **Attributi dell'oggetto Requisito**

I requisiti sono mantenuti in un database unico, in quanto un medesimo requisito potrebbe impattare più aree applicative.

L'accesso in scrittura al DB è controllato, e viene mantenuto lo storico delle modifiche del singolo requisito. In lettura, invece, i requisiti di un progetto sono, di norma, visibili e ricercabili da parte dell'intero team, che lavora sul progetto e ai suoi Utenti (Stakeholders).

Come sopra detto il requisito è l'elemento base per la gestione del processo di sviluppo. Si ritiene quindi utile definirne in modo dettagliato la sua struttura specificandone i suoi attributi nella seguente tabella.

ID	Identificativo univoco (assegnato automaticamente)
Versione	Cambia se il contenuto (descrizione) del requisito è modificato su richiesta dell'utente. (default: V0)
Titolo	
Descrizione	Descrive il requisito.
Tipologia	Classe a cui appartiene il requisito (nel precedente paragrafo sono definite e descritte 4 tipologie di requisito: Funzionali, Vincoli Funzionali, Non Funzionali, e Regole di Sicurezza; a queste si aggiungono le Decisioni Architetture definite nel capitolo seguente)
Dominio Applicativo	Area funzionale in cui è incluso il requisito
Manifestato da	Utente (stakeholder) che ha manifestato il requisito
Costo	Valorizzato in Euro. Il costo associato con l'implementazione viene valutato in fase di analisi ed aggiornato in ogni step successivo
Referente	Nome del Referente responsabile
Piano	Data di apertura, Previsioni di: fine analisi, fine disegno, approvazione, rilascio
Priorità	Priorità concordata con l'Utente (Alta, Media, Bassa)
Stato	Aperto, In analisi, In disegno, Approvato, Rilasciato
Complessità	Rischio connesso con l'implementazione (Alto/Medio/Basso)
Dimensionamento	FP (Function Points), GGUU (Giorni Uomo) o ambedue se necessario
Assegnato a	Responsabile della attività corrente
Giustificazione	Informazioni aggiuntive da parte dell'Utente che ha aperto il requisito
Test Cases collegati	Id.s dei test che verificano il requisito
Modelli Collegati	Vedi Sotto

Gli attributi descritti sono valorizzati inizialmente alla nascita del requisito e successivamente modificati. In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito
Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

### **Modelli usati nella descrizione del Requisito**

Per descrivere compiutamente le 4 tipologie di Requisito sopra definite occorrono dei modelli, sinteticamente descritti in allegato ed ampiamente trattati in letteratura. Di seguito si fornisce specifica indicazione di quali modelli si ritengono mandatori e opzionali per le diverse tipologie di requisito.

Link	Scenario Funzionale	Vincolo Funzionale	Requisito non-funzionale e di sicurezza	Decisione Architettonica (*)
Casi d'Uso	Wm		Wm	
Diagramma di Sequenza	Wm (alt)	Wo		Ro
Activity Diagram	Wm (alt)			Ro
Process Diagram	Wm (alt)	Wm		Ro
Modello E/R Deployment Diagram	Wm	Wm	Wm	Rm

Legenda: Wm = prodotto obbligatorio Wo = prodotto opzionale R(x) = Da altra Fase

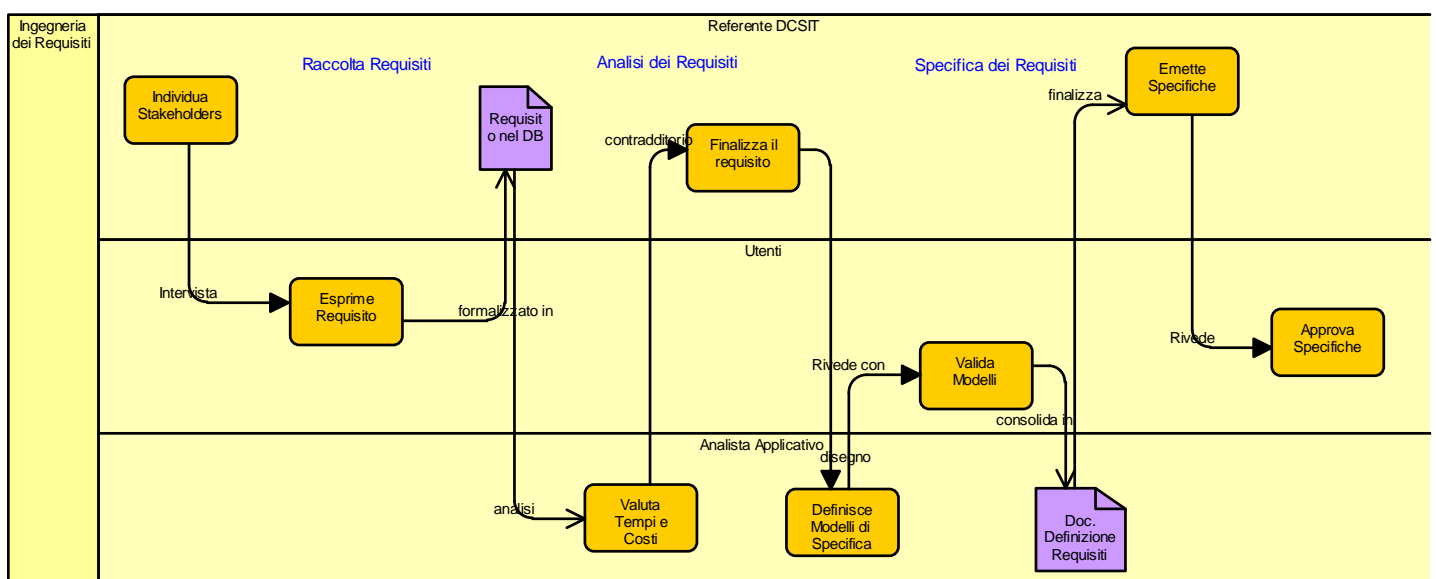
(\*) Le Decisioni Architettoniche vengono prodotte nella fase di Disegno (vedi)

## Attività

### Flusso Generale del processo

Il flusso generale è descritto nel seguente diagramma delle attività. Il diagramma fornisce le seguenti indicazioni:

- Le attività rappresentate dai rettangoli orizzontali in arancione
- Chi fa l'attività (attore) attraverso la suddivisione in fasce orizzontali
- I semilavorati (deliverables) specifici di questa fase rappresentati dai rettangoli verticali in viola.



### 1. Raccolta dei requisiti

- Obiettivo:** Questa fase mira ad identificare i requisiti che i vari “stakeholders” hanno sul dominio in analisi. Come precedentemente indicato le tipologie di requisito che dovranno essere esaminate sono 4:
  - Funzionali (requisiti da parte degli Utenti del Sistema)
  - Vincoli funzionali (le condizioni che devono essere rispettate nella soluzione).
  - Non Funzionali (ad esempio di performances).
  - Di Sicurezza e di Compliance (per policies aziendali e norme di legge).
- Output:** Apertura dei nuovi requisiti nel DB dei requisiti
- Attività:**
  - Identificazione degli Stakeholders.

- ii. Raccolta dei requisiti. I requisiti possono essere raccolti o con incontri diretti o mediante strumenti di collaborazione (anche del tipo “social networking” o “web 2.0”).
- iii. Formalizzazione dei Requisiti con gli Stakeholders, mediante la raccolta di documentazione o altro materiale (anche grafico ed al limite multimediale) a supporto della definizione del requisito.
- iv. Approvazione dei Requisiti da parte degli Stakeholders.

## 2. Analisi dei Requisiti.

- a. Obiettivo: Approfondimento del requisito, inclusa la definizione dei costi e rischi.
- b. Output: Dettaglio sufficiente da includere il requisito (o meno) nel piano di progetto
- c. Attività:
  - i. Approfondimento e classificazione del requisito
  - ii. Valutazione del costo di realizzazione
  - iii. Valutazione dei rischi
  - iv. Finalizzazione del requisito tra Referente ed Analista (in contraddittorio)

## 3. Specifica dei Requisiti:

- a. Obiettivo: I requisiti vengono analizzati a livello di specifica. In questa fase il requisito viene arricchito dai link agli artifacts collegati, ad. es. i diagrammi dei casi d'uso e quelli di sequenza che ne descrivono l'implementazione, i test cases che ne descrivono la modalità di verifica, ecc. In questa fase il requisito può essere oggetto a modifiche e raffinamenti da parte degli Utenti, e quindi può “riciclare” attraverso il processo di specifica.
- b. Output: Documento di Definizione dei Requisiti (l'indice prescritto per il documento è riportato di seguito ed è invece dettagliato nel documento A0-C03-0500-002, definito negli standard del Sistema di Qualità dell'Istituto).
- c. Attività:
  - i. Sviluppo dei modelli connessi col requisito
  - ii. Sviluppo del disegno logico dei dati (modello entity/relationship o modello delle classi di analisi)
  - iii. Rivisitazione del contenuto e conferma/modifica con l'Utente
  - iv. Rivisitazione dei costi e rischi
  - v. Aggiornamento del requisito nel DB
  - vi. Scrittura del Documento di Definizione dei Requisiti



## **Documento di Definizione dei Requisiti**

Di seguito si riporta l'indice del Documento Definizione dei Requisiti , mentre le regole di compilazione sono dettagliate nel documento A0-C03-0500-002, definito negli standard del Sistema di Qualità dell'Istituto.

### **INDICE**

- 1 INTRODUZIONE
  - 1.1 PREMESSA
  - 1.2 SCOPO DEL DOCUMENTO
  - 1.3 AREA DI APPLICAZIONE
  - 1.4 ABBREVIAZIONI
  - 1.5 DOCUMENTI CORRELATI
- 2 DEFINIZIONE DEL PROGETTO
  - 2.1 CONTESTO ATTUALE
  - 2.2 CONTESTO PREVISTO ED IPOTESI DI SOLUZIONE
  - 2.3 TIPOLOGIE DI UTENTI
- 3 MODELLO DEI REQUISITI
  - 3.1 ATTORI
  - 3.2 REQUISITI FUNZIONALI
  - 3.3 REQUISITI NON FUNZIONALI
  - 3.4 PROCESSI
- 4 ARCHITETTURA DEL SISTEMA
- 5 ANALISI DATI
- 6 PROTOTIPO
- 7 MODIFICHE IN CORSO D'OPERA
- 8 ALLEGATI

#### **4. Progettazione di alto livello dei test case**

- a. **Obiettivo:** è fondamentale che in fase di pianificazione si proceda alla identificazione dei test di "alto livello", ossia rappresentativi di un obiettivo di verifica/validazione che potrà corrispondere, una volta effettuata la progettazione di dettaglio (Design – specifiche di test), a molteplici condizioni di test. La progettazione di alto livello consente di avere una effettiva 'visione' dei test sulla base dei requisiti e del rischio, necessaria per analizzare la copertura dei requisiti stessi, valutare l'effort per le attività di test e procedere con una ottimizzazione dei test (rimozione dei test ridondanti), prima di proseguire con la progettazione di dettaglio. Essa è inoltre utile per disporre quanto prima di una completa visione dei requisiti stessi, oltre ad agevolare la loro revisione in presenza di eventuali ambiguità.
- b. **Output:** Documento "Tabella dei test (Progettazione Condizioni e casi di test di alto livello"



- c. Attività: i casi di test di alto livello sono identificati sulla base della scomposizione funzionale e dei requisiti approntata in fase di preparazione e analisi dei requisiti, eventualmente con strumenti di gestione dei requisiti.

## 2.2 Fase di Disegno Tecnico

In questa fase la documentazione del sistema passa da una descrizione di alto livello quasi interamente funzionale ad una serie di documenti tecnici di dettaglio. A tale scopo il sistema viene di solito suddiviso in Componenti, con interazioni minime e ben definite in fase di specifica. Un Componente è assegnato di norma ad uno ed un solo Designer Applicativo.

Obiettivo della fase è il rilascio del disegno completo del sistema, sia sotto l'aspetto complessivo, sia sotto l'aspetto del singolo componente.

Per fare ciò, oltre ai requisiti considerati nella fase di Ingegneria dei Requisiti, si dovranno prendere in considerazione le seguenti Scelte o Requisiti Architettonici:

- a. Gestione dei sistemi IT (supporto dei sistemi di gestione)
- b. Resilienza (capacità di continuare a funzionare, anche se in maniera degradata a fronte di problemi sull'ambiente)
- c. Flessibilità (facilità di implementare nuove funzioni)
- d. Portabilità (facilità di supportare un nuovo ambiente infrastrutturale)
- e. Manutenibilità (facilità nella manutenzione)
- f. Integrità dei Dati (utilizzo dei dati in maniera da non creare incongruenze anche in caso di Difetto)
- g. Tempi e Costi (a disposizione)
- h. Livello di Rischio (novità e rischio dei requisiti)
- i. IT Skills (novità della tecnologia nell'organizzazione)
- j. Infrastruttura esistente (tipo e possibilità dei sistemi esistenti)
- k. Stato dell'arte della tecnologia (allineamento con i trend di industria)
- l. Standards IT (rispetto degli standard interni ed esterni applicabili)

### **Obiettivi del Processo**

- Raffinare i modelli contenuti nelle specifiche fino a portarli ad essere una base sufficiente per la codifica.
- Prototipare le interfacce del sistema, rendendo possibile la verifica da parte degli utenti.
- Approfondire gli aspetti tecnici del sistema in sviluppo, onde renderlo realizzabile.
- Progettare i test di dettaglio (specifiche di test).

### **Risultati attesi**

Documentazione di Disegno ad un livello di dettaglio e coerenza tale da poter essere approvato come conforme a) alle Specifiche b) agli Standard dell'Istituto.

### **Uscita dalla Fase**

1. Le Specifiche Tecniche sono state approvate dalla DCSIT
2. Sono stati prodotti tutti i deliverables previsti (Modelli delle classi, Modello Fisico del DB, Manuali Utente, Disegno/Prototipo delle Interfacce, Test Cases di dettaglio).
3. I Costi e i piani di rilascio rivisti dopo il disegno sono approvati dal Management dell'INPDAP

### **Attori/Ruoli Aziendali coinvolti**

Referente Applicativo: Responsabile DCSIT del Progetto o Area Applicativa

Utenti (stakeholders): Rappresentanti delle linee operative che utilizzeranno il sistema

Architetto Applicativo: Figura tecnica specialista di sistemi software

Disegnatore: Figura tecnica specialista di sviluppo applicativo

DBA: Data Base Administrator

Disegnatore GUI: Figura tecnica specialista di sviluppo di interfacce.

### **Work products – input**

[PL] Principi, Linee Guida e Standards di riferimento

[UC] Use Case Model

[SQ] Sequence Diagram

[AD] Activity Diagram/Process Diagram

[IU] Specifiche interfacce utente

[ER] Modello concettuale dei dati

### **Output**

[AR] Diagramma Architettuale (Deployment)

[CL] Modello delle Classi e dei Componenti

[SQ] Sequence Diagram

[AD] Activity Diagram/Process Diagram

[CL] Disegno delle Classi/Oggetti

[UI] Interfacce utente

[RU] Indicazioni sul possibile riuso di componenti

[DB] Disegno fisico del DB

[MU] Manuali Utente

[TS] Dettaglio casi di Test (script) – Specifiche di Test

- [TF] Test Cases Funzionali
- [TP] Test Plan Funzionale
- [TI] Test Cases di Integrazione

### **Attributi dell'Oggetto Disegno (Specifiche Tecniche del Sistema o Disegno del Componente)**

Esistono due livelli dell'oggetto: quello di Sistema e quello di Componente, che hanno però gli stessi attributi di seguito specificati.

ID	Identificativo univoco (assegnato automaticamente)
Versione	Cambia per ogni modifica apportata dopo l'approvazione del documento di Disegno
Titolo	Nome del Sistema Applicativo o del singolo Componente oggetto del Disegno
Descrizione	Breve descrizione dello scopo funzionale del sistema o componente
Stato	Completato/Approvato/Modificato
Requisiti Collegati	Id. dei requisiti collegati
Test Collegati	Test Funzionali e di Integrazione collegati
Prototipo	Puntatore all'eventuale ambiente di prototipo
Modelli Collegati	Vedi sotto

In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito
Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

### **Modelli usati nelle Specifiche Tecniche**

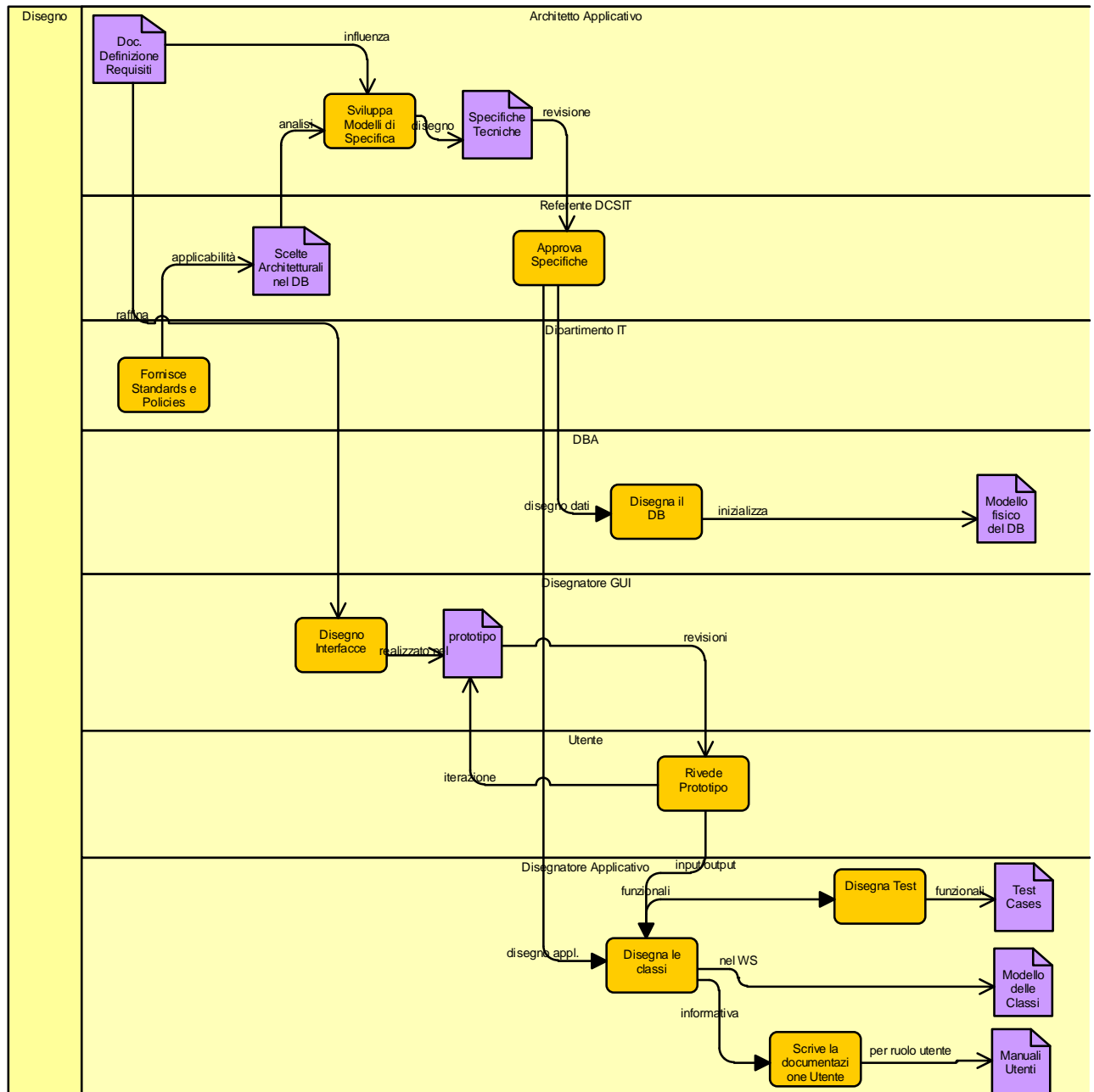
	Disegno complessivo del Sistema	Disegno del singolo Componente
Modello delle Classi		Wm
Modello E/R	Rm	
Sequence Diagrams	Wm	Wm
Modello dei Componenti	Wm	Rm
Activity Diagram		Wo
Interfacce Utente		Rm
Disegno fisico del DB	Wm	
Diagramma Architeturale (Deployment)	Wm	

Legenda: Wm = prodotto obbligatorio Wo = prodotto opzionale R(x) = da altra fase

## Attività

### Flusso Generale del processo

(per il formalismo si rimanda al capitolo precedente)



Di seguito, le attività dalla 1 alla 6 fanno parte del Disegno del Sistema, mentre la 7 attiene al Disegno dei Componenti.

### 1. Realizzazione delle Specifiche Tecniche del Sistema:

- a. Obiettivo: Questa fase mira a complementare il contenuto delle Specifiche con le Scelte Architetturelle della DCSIT.
- b. Output Documento di Specifiche Tecniche.
- c. Attività:
  - i. Raccolta e formalizzazione delle Scelte Architetturelle come Requisiti nel DB dei Requisiti;
  - ii. Valutazione degli elementi riusabili (assets) nel catalogo interno o disponibili esternamente
  - iii. Sviluppo dei modelli richiesti dalle Specifiche Tecniche;
  - iv. Scrittura delle Specifiche Tecniche;
  - v. Produzione della proposta di assetizzazione (valutazione costi e tempi per la trasformazione di componenti prodotti nel progetto e potenzialmente riusabili, in asset da inserire nel catalogo interno)
  - vi. Approvazione della DCSIT.

### 2. Disegno del DB

- a. Obiettivo: Questa fase ha lo scopo di passare dal disegno logico dei dati ad il disegno fisico del DB opportunamente ottimizzato.
- b. Output Definizione delle tabelle e degli spazi da allocare.
- c. Attività:
  - i. Revisione del Modello dei dati e dei carichi di lavoro;
  - ii. Revisione delle esigenze informative dei Componenti (query);
  - iii. Normalizzazione del modello;
  - iv. De-normalizzazione delle tabelle critiche.

### 3. Disegno e Prototipo delle Interfacce:

- a. Obiettivo: Questa fase raffina progressivamente i contenuti delle Specifiche fino ad arrivare ad una definizione delle interfacce (utente) di dettaglio sufficiente ad avere l'approvazione dell'Utente. L'attività è altamente iterativa e prevede numerosi raffinamenti successivi del prototipo
- b. Output: Rappresentazione HTML o XML dei pannelli ed eventuale ambiente di prototipazione (sviluppo e run).
- c. Attività:

- i. Definizione degli elementi grafici e standard comuni;
- ii. Codifica del markup delle interfacce (inclusi i messaggi scambiati con altri sistemi);
- iii. Realizzazione dell'eventuale prototipo;
- iv. Verifica con L'Utente.

#### 4. **Disegno delle classi:**

- a. Obiettivo: Questa fase ha lo scopo di affinare progressivamente i modelli delle Classi/Oggetti contenuti nelle specifiche ed assegnati allo specifico Sviluppatore fino a portarli ad essere una buona base per la codifica delle stesse.
- b. Output della fase: il disegno di tutte le classi del componente nel workspace del tool IDE. (Integrated Development Environment)
- c. Attività:
  - i. Applicazione dei framework applicativi appropriati al caso, o decisi in fase di specifica ;
  - ii. Introduzione dei pattern di disegno e delle best practices;
  - iii. Modellazione grafica UML2 delle Classi.

#### 5. **Produzione della Documentazione Utente**

- a. Obiettivo: Questa fase produce il draft dei manuali Utente che sarà testato nei test di Sistema ed Utente
- b. Output: Un manuale per ciascuna tipologia di Utente (ad es. Amministratore di sistema, Addetto allo sportello, Responsabile del processo).
- c. Attività:
  - i. Scrittura dei manuali in formato "Bozza"
  - ii. Revisione Interna
  - iii. Eventuale test congiunto col prototipo

#### 6. **Scrittura del piano di Test Funzionale**

- a. Obiettivo: Questa fase pianifica il Test Funzionale che avverrà nella fase di Costruzione
- b. Output: Piano di Test e Test Cases funzionali. Alcuni Test Cases sono inoltre designati come "Test di Integrazione".
- c. Attività:
  - i. Definizione dei principi e linee guida.
  - ii. Definizione della matrice di Copertura (tracciabilità) dei Requisiti (si completa la tabella dei test).





- iii. Specifiche per la preparazione dell'ambiente e dei dati.
- iv. Progettazione dei casi di test di dettaglio (vedi specifiche dell'Oggetto Test Case).
- v. Identificazione dei Casi di Test di integrazione (da eseguire ad ogni nuovo build).

## 2.3 Fase di Costruzione

In questa fase il sistema passa da uno stato puramente concettuale ad uno fisico, eseguibile su un elaboratore di dati. Il lavoro in questa fase è altamente parallelizzato ed a tale scopo si prosegue l'approccio per Componenti usato nella fase di Disegno. Un Componente è assegnato di norma ad uno ed un solo Sviluppatore Applicativo.

Obiettivo della fase è il rilascio alle attività di test dei moduli eseguibili di buona qualità e allineati con i modelli ed il codice di un Workspace IDE in maniera tale da assicurarne la facile manutenzione correttiva ed evolutiva del Componente stesso.

### **Obiettivi del Processo**

- Costruire un ambiente integrato tra disegno e codice, che consenta di effettuare le modifiche manutentive al corretto livello di astrazione
- Testare il codice prodotto in un ambiente dedicato ed in maniera autonoma
- Costruire gli eseguibili necessari al deployment negli ambienti di test

### **Risultati attesi**

Workspace allineata con gli Eseguibili consegnati al gruppo di Test.

### **Uscita dalla Fase**

1. I casi di test funzionali e di integrazione girano senza errori
2. Le workspaces di sviluppo sono state messe in configurazione

### **Attori/Ruoli Aziendali coinvolti**

Sviluppatore: Figura tecnica specialista di sviluppo applicativo

### **Work products – input**

- [PL] Principi, Linee Guida e Standards di riferimento
- [UC] Use Case Model
- [SQ] Sequence Diagram
- [AD] Activity Diagram/Process Diagram
- [IU] Specifiche interfacce utente
- [AR] Diagramma Architeturale (Deployment)
- [CL] Modello delle Classi e dei Componenti
- [RU] Indicazioni sul possibile riuso di componenti
- [ER] Modello concettuale dei dati
- [DB] Disegno fisico del DB

[CL] Disegno delle Classi/Oggetti (nel Workspace IDE)

[UI] Interfacce utente

[PR] WBS e Gantt del progetto

[TF] Test Cases Funzionali

[TP] Test Plan Funzionale

[TI] Test Cases di Integrazione

### **Output**

[CO] Codice delle Classi/Oggetti

[UT] Ambiente di Unit Test

[DU] Eseguibili per il deployment

[ST] Test Script automatici

[RT] Risultati dei Test

[PD] Problem / Defect Log

[CF] DB di Configuration Management

### **Attributi dell'oggetto Componente Applicativo**

Questo oggetto descrive il codice di un componente nelle sue versioni sorgente ed eseguibile, così come contenuta nella workspace del tool IDE dello sviluppatore applicativo

ID	Identificativo univoco (assegnato automaticamente)
Versione	Cambia ogni qualvolta un nuovo eseguibile è passato alla Fase di Test. (default: V0)
Titolo	
Descrizione	Breve descrizione delle modifiche contenute nella versione del componente
Contenuto	Elenco delle Classi (oggetti)
Stato	In sviluppo/In manutenzione
Casi di Test collegati	Funzionali e di Integrazione
Requisiti collegati	Requisiti implementati
Fix collegate	Modifiche di soluzione di problemi contenute
Eseguibili	Lista degli eseguibili risultanti (la versione è la medesima del Componente)
Modelli Collegati	Vedi sotto

In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito

Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

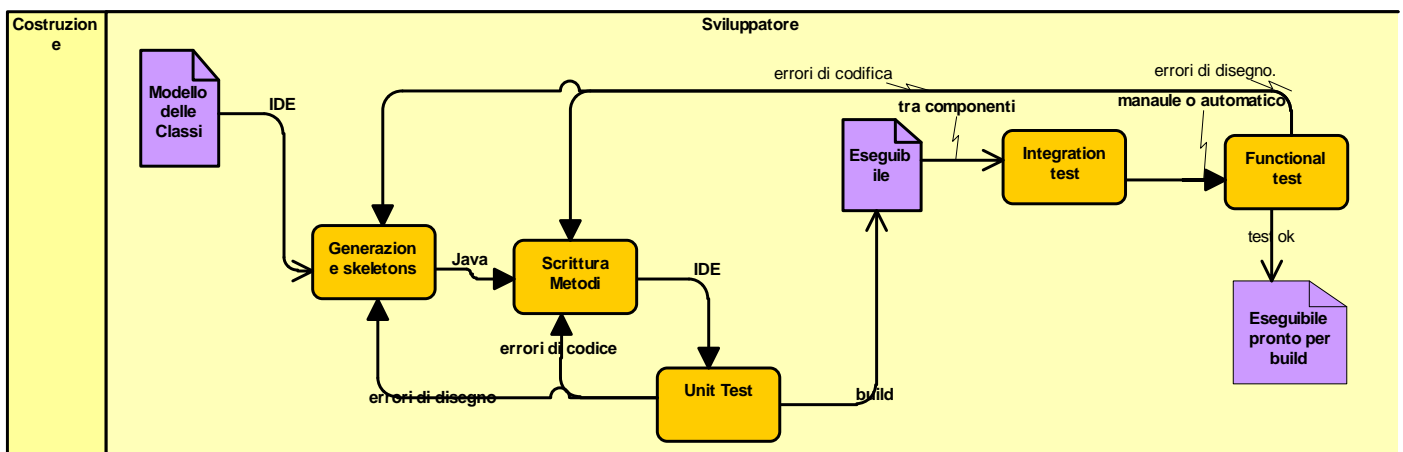
### Modelli usati nella fase

	Componente
Use Case	Rm
Modello delle Classi	Wm
Diagrammi di Sequenza	Ro
Process Diagram	Ro
Diagramma di Deployment	Rm
Modello dei Componenti	Rm
Activity Diagram	Ro

### Attività

#### Flusso Generale del processo

Figura 1. Flusso Generale



#### 1. Generazione Skeletons

- a. Obiettivo: Questa fase mira a generare lo scheletro delle classi a partire dai modelli contenuti nel Workspace dell'IDE da cui sono stati prodotti i documenti di Disegno
- b. Output Skeletons delle Classi.
- c. Attività:
  - i. Generazione degli skeletons
  - ii. Unit Test di integrazione

#### 2. Codifica.

- a. Obiettivo: Una volta che lo scheletro della classe è stato generato, si passa alla scrittura del codice dei metodi.
- b. Output: Tutti i metodi presenti nel workspace sono stati scritti, anche se la fase successiva di Unit Test può iniziare anche con alcuni dei metodi ancora vuoti.
- c. Attività:
  - i. Codifica dei Metodi
  - ii. Generazione degli eseguibili

### 3. Unit Test:

- a. Obiettivo: Una volta che una classe, o i suoi metodi più importanti, siano codificati, l'ambiente IDE offre gli strumenti per eseguirli in un ambiente simulato.
- b. Output: il workspace IDE con tutti gli errori corretti in grado di generare l'eseguibile per il test.
- c. Attività:
  - i. Preparazione degli scaffolding (simulatori) e dei dati di test
  - ii. Esecuzione di white-box test (di norma non formalizzati)
  - iii. Rilevazione degli errori e loro correzione (sul modello di disegno o sul codice a seconda del tipo)
  - iv. Rigenerazione degli eseguibili
  - v. Ripetizione dei test.

### 4. Generazione degli Eseguibili per il Test:

- a. Obiettivo: Questa fase del processo consiste nel passaggio dell'applicazione al test
- b. Output: Salvataggio della workspace in configurazione e passaggio del build in test.
- c. Attività:
  - i. Generazione degli Eseguibili da parte degli sviluppatori
  - ii. Salvataggio del Workspace sotto la Gestione della Configurazione
  - iii. Preparazione del rapporto di esecuzione degli UT e di rimozione degli errori
  - iv. Generazione e documentazione del build dell'applicazione
  - v. Installazione del build nell'ambiente di test.

### 5. Esecuzione del Test di Integrazione e Funzionale:

- a. Obiettivo: Verificare l'integrazione del nuovo eseguibile del componente e le funzioni rilasciate.
- b. Output: Tutti i casi di test eseguono senza errori.
- c. Attività:

- i. Preparazione dell'ambiente per i test funzionali
- ii. Esecuzione dei test cases di Integrazione
- iii. Esecuzione dei test cases funzionali con l'utilizzo dei manuali Utente
- iv. Rimozione degli errori riscontrati
- v. Preparazione del rapporto di esecuzione e di rimozione degli errori.

## 2.4 Avviamento in Esercizio

La Fase di Avviamento (Messa in Esercizio) raccoglie tutte le attività connesse al dispiegamento controllato del software applicativo sui suoi ambienti di esecuzione temporanea (Test di Sistema, Test Prestazionale, Test di Accettazione Utente) e definitiva (Esercizio, Business Continuity, Verifiche di Manutenzione). In questi ambienti il sistema viene dispiegato in maniera completa (senza alcuna funzione simulata) e con tutti i prodotti prerequisites al medesimo livello. Operativamente il lavoro è diviso tra vari team (Team di Test, Team di Sviluppo per il fixing dei problemi, Team Sistemistico per la preparazione e la gestione degli ambienti) ed è quindi essenziale il buon funzionamento dei meccanismi e processi di collaborazione, controllo, suddivisione ed assegnazione della attività (onde evitare problemi di contaminazione della baseline).

### **Obiettivi del Processo**

Il processo è teso a garantire un sistema complessivo (Hardware, Software di Base, Software prerequisite, Rilascio Applicativo, Documentazione) autoconsistente ed in grado di supportare senza errori i requisiti previsti.

### **Risultati attesi**

Tutti gli elementi del sistema sono installati ed operativi al livello specificato in fase di disegno.

### **Uscita dalla Fase**

1. Tutti i livelli di test previsti sono stati completati (i Test Cases girano senza evidenziare anomalie)
2. Le attività di Messa in Esercizio sono completate
3. E' possibile effettuare la manutenzione

### **Attori/Ruoli Aziendali coinvolti**

Gestore della Configurazione: Figura Tecnica esperta negli strumenti di configurazione e build

Sviluppatori: Figura tecnica specialista di sviluppo applicativo

Architetto del Test: progetta in collaborazione con il Referente e gli Utenti le specifiche dei test prestazionali

Capo Progetto: contribuisce a stabilire la pianificazione dei test prestazionali

Test Designer: realizza le specifiche tecniche dei test cases prestazionali

Referente : Responsabile del Progetto o Area Applicativa

Test Executor: figura tecnica che esegue i test cases prestazionali

Utenti (stakeholders): Rappresentanti delle linee operative che utilizzeranno il sistema ed eseguono i casi di test di accettazione del sistema

**Sistemisti:** Figura tecnica specialista di gestione dei sistemi IT, dei prodotti middleware e dei Database

**Work products – input**

[PL] Principi, Linee Guida e Standard di riferimento

[RQ] DB dei Requisiti

[AR] Diagramma Architetture (Deployment)

[PR] WBS e Gantt del progetto

[CF] DB di Configuration Management

**Output**

[IP] Procedura e checklist di Installazione

[SY] Ambienti Operativi funzionanti

[PD] Problem / Defect Log

[CF] DB di Configuration Management

**Attributi dell'oggetto Sistema**

ID	Identificativo univoco (assegnato automaticamente)		
Versione	Identifica il rilascio di un prodotto. È nel formato Versione,Rilascio,Livello di Modifica (es: 3.1.2). Il Livello di Modifica è normalmente associato al rilascio di correzioni di difetti, il Rilascio all'inclusione di nuovi requisiti, la Versione alla presenza di modifiche sostanziali al contenuto o all'ambiente.		
Descrizione	Elenca le modifiche incluse nel rilascio in oggetto Le modifiche possono essere tecnico/funzionali (rilascio di requisiti), o correzioni.		
Include	Elenca gli eseguibili inclusi ed il loro livello		
Stato	Planned/Installed/Operative		
Situazione negli ambienti	Lista per ogni ambiente target,lo stato e la data relativa ad es:		
	System test	Operative	Dal 3 Marzo 2009
	Performance Test	Installed	Dal 6 Marzo 2009
	Esercizio	Planned	Per 2 Dicembre 2009
Procedura di Build	Descrive i passi per effettuare il build manuale o punta allo script automatizzato		
Procedura di Installazione	Descrive i passi per effettuare l'installazione manuale o punta allo script automatizzato		
Procedura di Verifica	Descrive i passi per effettuare la verifica manuale di corretta installazione o punta allo script automatizzato		
Ambiente di esecuzione	Link ai diagrammi di deployment (in configurazione) che descrivono le specifiche di ognuno degli ambienti di esecuzione		
Modelli Collegati	(vedi sotto)		



In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito
Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

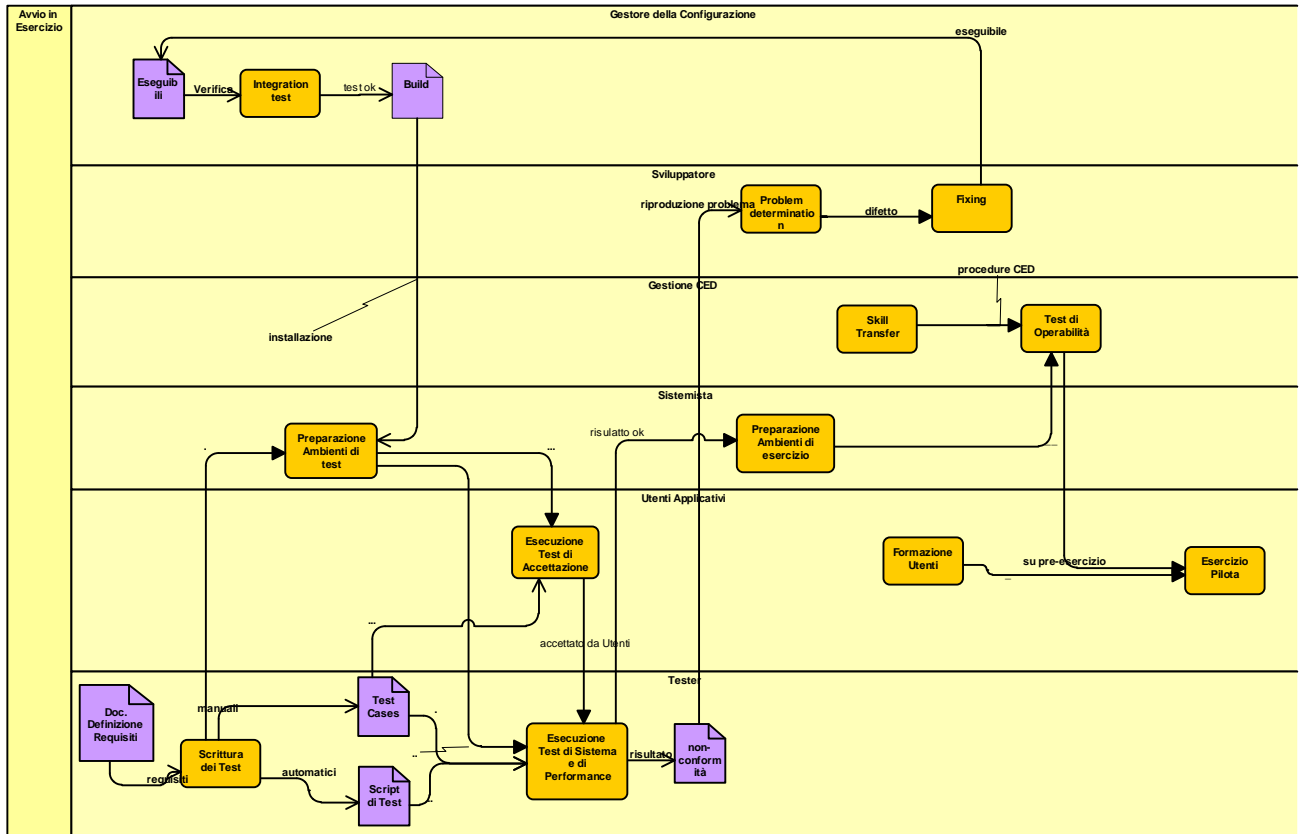
### **Modelli usati per ambiente di deployment**

Nella tabella sono correlati i modelli con gli ambienti di rilascio per l'esecuzione temporanea e definitiva.

	Sytem Test, Test Prestazionale	User Acceptance Test	Esercizio, Bus. Continuity	Test di Manutenzione
Diagramma di Processo	Wm	Wm		Rm
Diagramma di Attività	Wm	Wm		Rm
Diagramma di Deployment	Wm	Wm	Wm	Wm
Modello dei Componenti	Rm	Rm	Rm	Rm

## Attività

### Flusso Generale del processo



#### 1. Scrittura del Test Plan e dei Test Cases

- a. **Obiettivo:** Questa fase pianifica la realizzazione dei Test (di Sistema, di Performance, di Accettazione, di Operabilità) che avverranno nella Messa in Esercizio (ma che sono stati progettati nelle fasi precedenti)
- b. **Output** Piano di Test e Test Cases. Alcuni Test Cases sono inoltre designati come "Test di Accettazione Utente".
- c. **Attività:**
  - i. Definizione della matrice di Copertura (tracciabilità) dei Requisiti
  - ii. Specifiche per la preparazione dell'ambiente di test e dei dati
  - iii. Scrittura del singolo Caso di Test (vedi specifiche dell'Oggetto Test Case)

#### 2. Build

- a. **Obiettivo:** Il build costruisce un livello specificato dell'eseguibile dell'applicazione a partire dai livelli appropriati dei moduli eseguibili dei componenti.

- b. Output: Una configurazione in grado di passare con successo i casi di test dell'Integration Test eseguiti dal gruppo di sviluppo.
- c. Attività:
  - i. Raccolta degli eseguibili, dei sorgenti e delle procedure di build.
  - ii. Esecuzione del Build e baselining della configurazione.
  - iii. Installazione del Build in ambiente di Integration Test.
  - iv. Esecuzione dell'Integration Test.
  - v. (eventuale riciclo in caso di errori).

### 3. Preparazione degli ambienti e degli script di Test

- a. Obiettivo: Questa fase produce gli strumenti per i Test (di Sistema, di Performance, di Accettazione, di Operabilità) che avverranno nella Messa in Esercizio.
- b. Output Ambienti fisici con installati i giusti prerequisiti software, Script per gli strumenti di esecuzione automatizzata dei test.
- c. Attività:
  - i. Installazione prerequisiti software
  - ii. Preparazione dei dati di test
  - iii. Installazione strumenti di test
  - iv. Scrittura degli Script
  - v. Installazione del Build applicativo

### 4. System Test

- a. Obiettivo: Verificare il sistema in ambiente simile a quello di produzione. Questo include la ri-esecuzione di alcuni specifici Test Funzionali (scelti in base alla loro importanza nella verifica degli obiettivi del sistema) e l'esecuzione di Test specificatamente orientati al test dei Requisiti non-Funzionali e delle Scelte Architetture, in special modo quelli di Sicurezza.
- b. Output: Tutti i casi di test girano senza produrre errori
- c. Attività:
  - i. Esecuzione dei Test
  - ii. Registrazione degli Errori
  - iii. Problem determination e rimozione delle cause di Difetto
  - iv. Installazione della nuova configurazione
  - v. Test di regressione del problema
  - vi. (eventuale riciclo in caso di errori)

## 5. Performance Test

- a. Obiettivo: Verificare i requisiti di capacity e di performance del sistema mediante tools simulatori di carico (utenti concorrenti e traffico)
- b. Output: E' verificato il raggiungimento degli obiettivi quantitativi
- c. Attività:
  - i. Esecuzione dei Test
  - ii. Registrazione degli Errori
  - iii. Problem determination e rimozione delle cause di non-conformità
  - iv. Installazione della nuova configurazione
  - v. Test di regressione del problema
  - vi. (eventuale riciclo in caso di non raggiungimento dei target)

## 6. User Acceptance Test

- a. Obiettivo: Verificare, mediante un test effettuato in contraddittorio, che il sistema sia conforme ai requisiti Utente approvati nelle Specifiche.
- b. Output: Tutti i test previsti sono stati eseguiti e le non-conformità rilevate (incluse quelle della documentazione) sono state rimosse
- c. Attività:
  - i. Installazione prerequisiti software nell'ambiente di pre-esercizio
  - ii. Preparazione dei dati di test
  - iii. Esecuzione dei Test
  - iv. Registrazione delle non-conformità
  - v. Problem determination e rimozione delle cause di non-conformità
  - vi. Installazione della nuova configurazione
  - vii. Test di regressione del problema

## 7. Messa in Esercizio

- a. Obiettivo: Instalare e verificare l'ambiente di Esercizio, Formazione, e quello di Business Continuity.
- b. Output: Il sistema entra in fase operativa
- c. Attività:
  - i. Installazione prerequisiti software
  - ii. Esecuzione e test configurazione di rete
  - iii. Integrazione negli ambienti di System e Network Management



- iv. Skill transfer agli operatori e sistemisti CED
- v. Test di operabilità del sistema secondo le procedure e gli standard dell'esercizio
- vi. Formazione degli Utenti
- vii. Esercizio controllato sulle sedi pilota
- viii. Cut-off in esercizio

## 2.5 Manutenzione

Le attività di Manutenzione iniziano quando un componente viene rilasciato al test e proseguono fino alla dismissione dell'applicazione.

L'attività consiste nel registrare un errore del sistema, nel determinarne il difetto che ne è la causa e nel rimuoverlo mediante una fix del disegno e/o del codice.

La caratteristica di queste attività è quella di non poter essere pianificate e di essere disperse nel tempo, quindi è essenziale per il loro svolgimento il supporto di uno strumento che ne consenta il controllo ordinato anche a livello "storico".

### **Obiettivi del Processo**

Il processo è teso a prendere in carico un Difetto avvenuto in test o in esercizio e a provvedere e verificare la relativa correzione.

### **Risultati attesi**

Il Difetto non si ripresenta più nel sistema modificato.

### **Uscita dalla Fase**

L'applicazione è dismessa dall'esercizio (fine vita).

### **Attori/Ruoli Aziendali coinvolti**

Sviluppatore: Figura tecnica specialista di sviluppo applicativo

Tester: Figura tecnica esperta nella riproduzione di condizioni di Difetto in un ambiente controllato (di test)

Gestore della Configurazione: Figura Tecnica esperta negli strumenti di configurazione e build

Help Desk: Figura tecnica esperta in supporto alle procedure utente

Sistemista: Figura Tecnica esperta di sistemi operativi, middleware e procedure di installazione

### **Work products – input**

[PL] Principi, Linee Guida e Standards di riferimento

[UC] Use Case Model

[SQ] Sequence Diagram

[AD] Activity Diagram/Process Diagram

[IU] Specifiche interfacce utente

[AR] Diagramma Architettuale (Deployment)

[CL] Modello delle Classi e dei Componenti

[DB] Disegno fisico del DB

[CL] Disegno delle Classi/Oggetti (nel Workspace IDE)

[UI] Interfacce utente

[CO] Codice delle Classi/Oggetti

[UT] Ambiente di Unit Test

[DU] Eseguibili per il deployment

[PD] Problem / Defect Log

[CF] DB di Configuration Management

### **Output**

[CF] DB di Configuration Management

[PD] Problem / Defect Log

### **Attributi dell'oggetto Difetto/Fix**

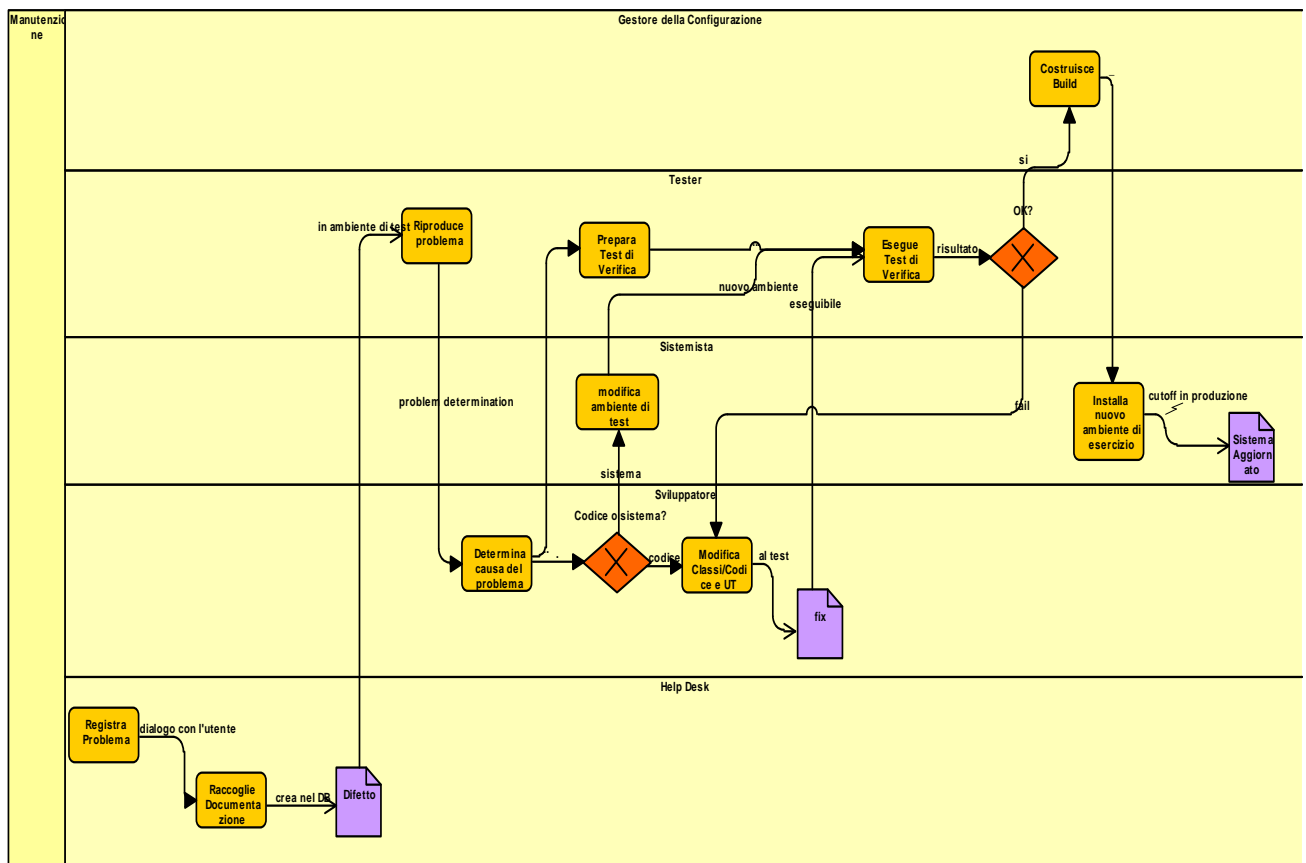
ID	Identificativo univoco (assegnato automaticamente)
Versione dell'Applicazione	Identifica il rilascio dell'applicazione su cui è stato riscontrato l'errore
Descrizione	Descrive quanto necessario per ricreare la condizione di errore
Tipologia	Test di Sistema/Test Prestazionale/Test Utente/Esercizio/Business Continuity/Manutenzione
Fix	Descrive l'origine del problema e la maniera per correggerlo
Creato da	Persona che ha riscontrato il problema
Data creazione	Data di identificazione dell'errore
Piano di Chiusura	Data prevista/effettiva (dipende dallo stato) per la chiusura
Stato	In Analisi/In sviluppo/In test/Chiuso/Non riproducibile
Include	Elenca gli eseguibili inclusi nella fix ed il loro livello
Procedura di Build	Descrive i passi per effettuare il build manuale o punta allo script automatizzato
Procedura di Installazione	Descrive i passi per effettuare l'installazione manuale o punta allo script automatizzato
Procedura di Verifica	Descrive i passi per effettuare la verifica manuale o punta allo script automatizzato
Modelli Collegati	(vedi sotto)

Modelli usati nella fase

	Difetto/Fix
Use Case	Rm
Modello delle Classi	Wm
Diagrammi di Sequenza	Ro
Process Diagram	Ro
Diagramma di Deployment	Rm
Modello dei Componenti	Ro
Activity Diagram	Ro

## Attività

### Flusso Generale del processo



#### 1. Problem Recording

- Obiettivo:** Catturare sufficienti informazioni sul Difetto in modo da permetterne la riproduzione in un ambiente controllato di test.
- Output:** Problema aperto nello strumento di configurazione



c. Attività:

- i. Ricerca sul DB dei difetti se l'errore si è già manifestato (regressione)
- ii. Raccolta della documentazione
- iii. Apertura del problema
- iv. Eventuale interazione (domanda/risposta) con lo sviluppatore

**2. Problem Determination**

- a. Obiettivo: Riprodurre il problema in ambiente di test e determinarne la causa.
- b. Output: Aggiornamento del problema
- c. Attività:
  - i. Studio della documentazione fornita
  - ii. Creazione dell' ambiente e dei dati di contorno
  - iii. Riproduzione del problema
  - iv. Analisi della causa del problema
  - v. Creare un nuovo test di regressione che permetta di verificare che il nuovo sistema esegue correttamente

**3. Fixing e Verifica**

- a. Obiettivo: Rimuovere le cause del Difetto.
- b. Output: il workspace IDE con tutti l'Difetto corretti in grado di generare l'eseguibile per il test.
- c. Attività:
  - i. Ricercare se sono state già create fix per problemi analoghi;
  - ii. Modificare il modello delle classi o il codice dei componenti in Difetto;
  - iii. Generare i nuovi eseguibili;
  - iv. Passare gli eseguibili al Gestore della Configurazione per eseguire il build;
  - v. Eseguire il nuovo test di regressione;
  - vi. Chiudere il problema.

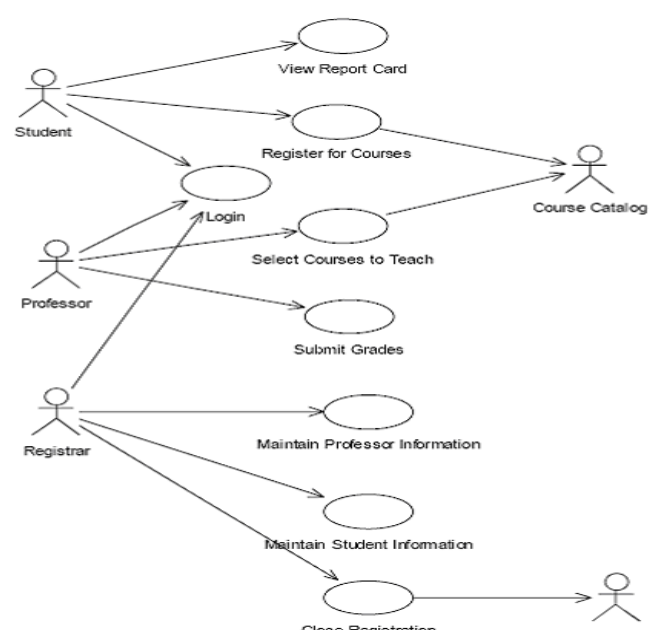
## 3 Modelli

### 3.1 Use Case

Lo Use Case è il modello di base per descrivere uno scenario funzionale .

E' basato sulla notazione UML2, e ha come elementi:

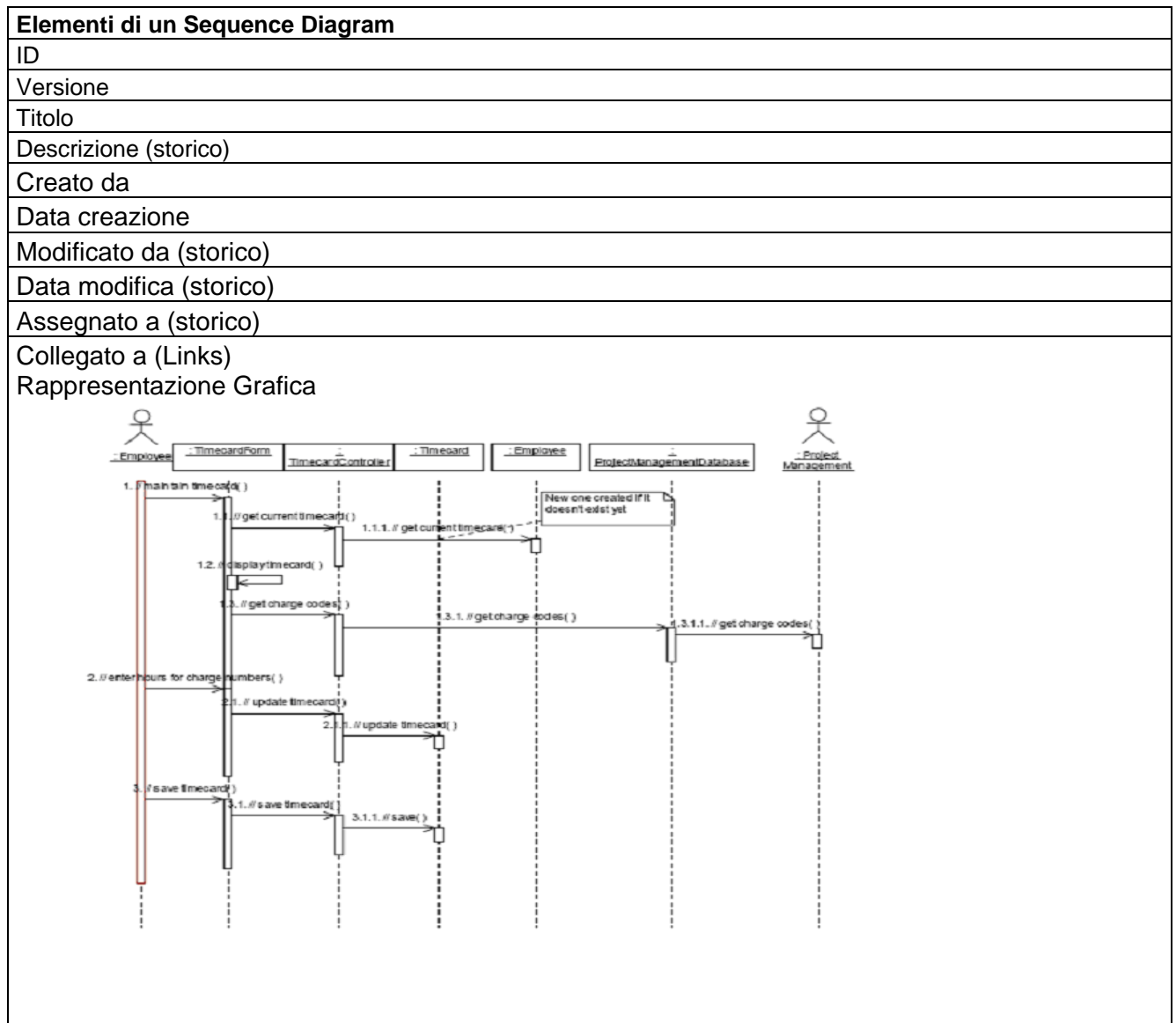
- Uno o più Attori, vale a dire Utenti del sistema (N.B. questi possono essere Ruoli Utente o Sistemi Esterni che si interfacciano con quello in sviluppo)
- Gli Use Cases di una certa Area. Lo Use Case è una interazione col sistema che lascia lo stesso in uno stato autoconsistente.

Elementi di uno Use Case
ID
Versione
Titolo
Descrizione (storico)
Creato da
Data creazione
Modificato da (storico)
Data modifica (storico)
Complessità
Assegnato a (storico)
Collegato a (Links)

Rappresentazione Grafica

### 3.2 Sequence Diagram

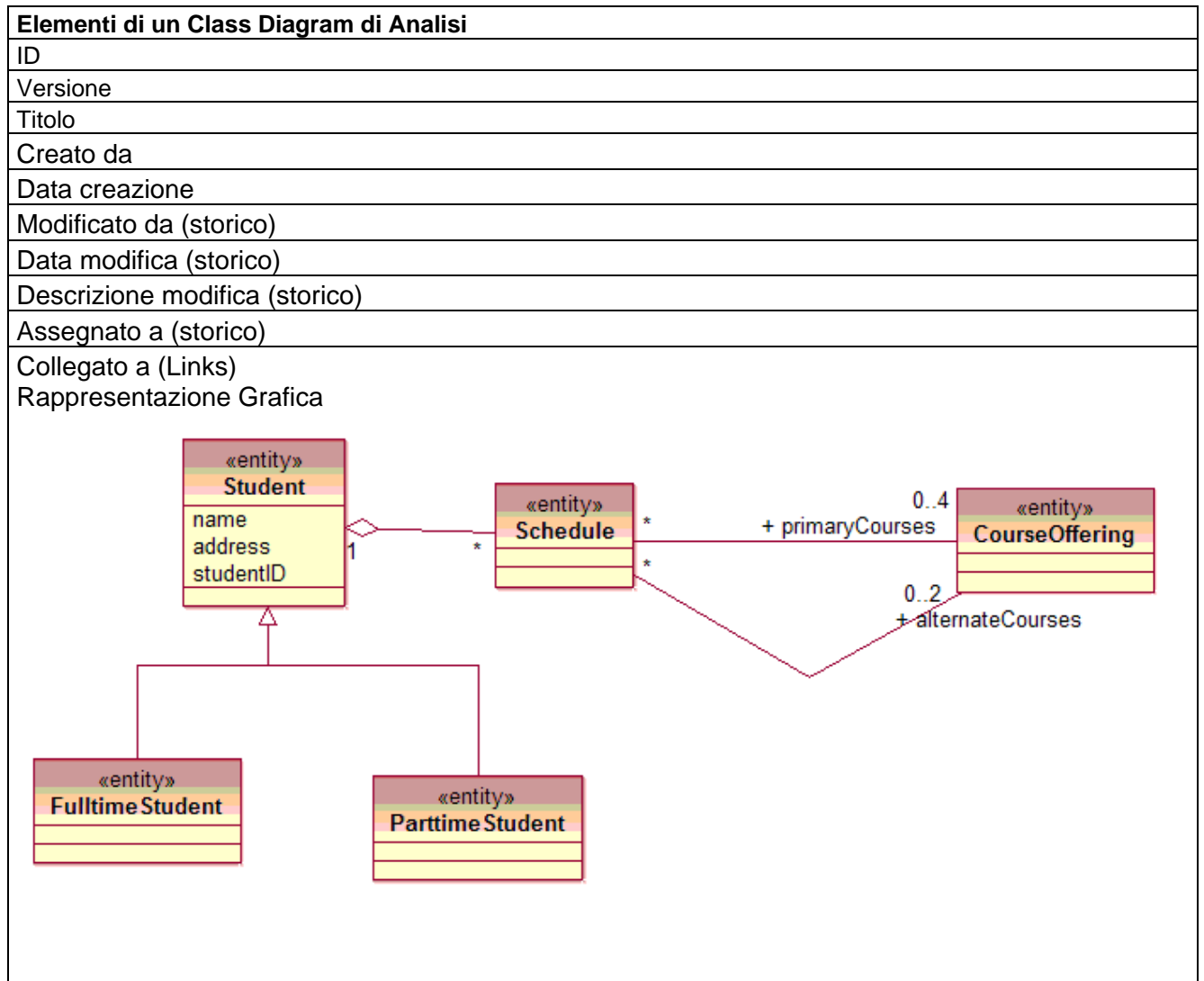
Il Sequence Diagram usualmente completa un altro diagramma di tipo statico. Questo diagramma mostra:

- Come Uno o più Attori interagiscono col sistema.
- Come il Sistema complementa uno Use Case, dei Componenti o delle Classi del sistema nel caso sia volto a spiegarne il funzionamento interno, i nodi o gli oggetti di deployment nel caso sia volto a spiegare il ruolo e le interazioni tra gli elementi dell'infrastruttura fisica.



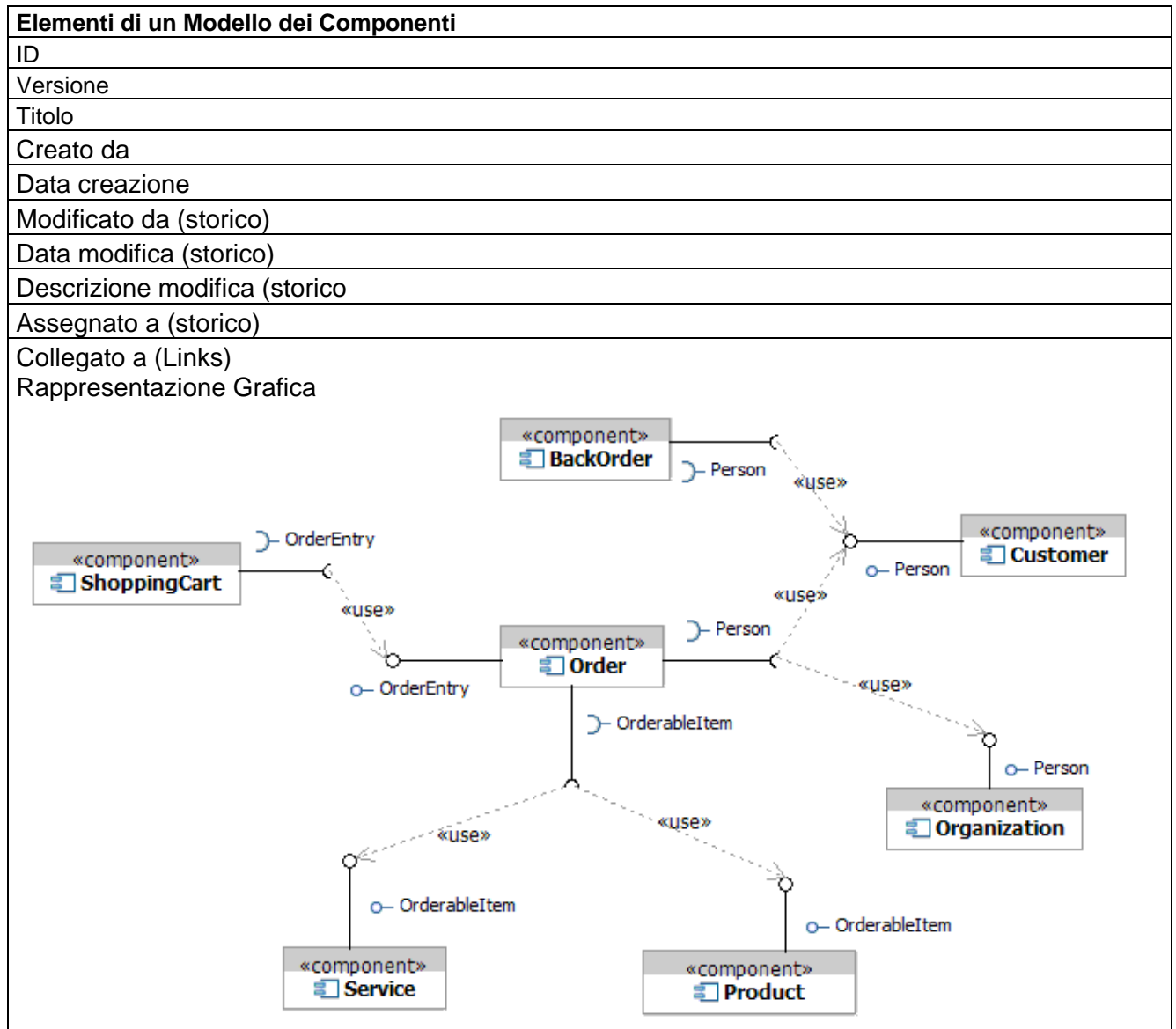
### 3.3 Diagramma Entity-Relationship o di Classi di Analisi

Questo diagramma è usato per rappresentare le entità logiche che costituiscono il dominio applicativo (inclusi i loro attributi) e le loro relazioni.



### 3.4 Modello dei Componenti

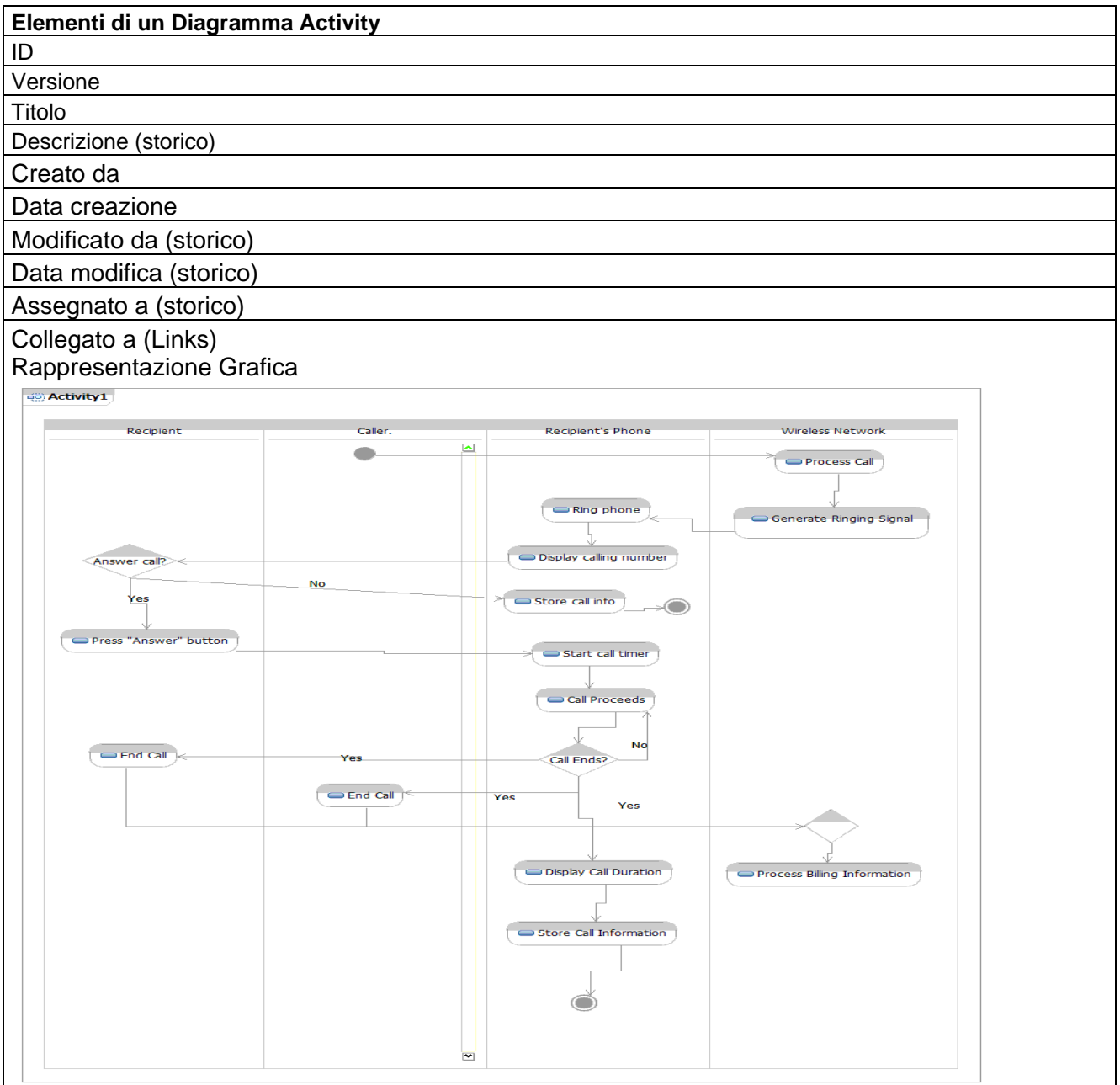
Il modello dei Componenti rappresenta i macro-blocchi dell'applicazione e le loro relazioni. I Componenti hanno la caratteristica di dialogare solo attraverso le loro interfacce, e sono quindi una best practice per la strutturazione di un sistema in elementi di ordine e complessità inferiore.



### 3.5 Activity Diagram

Anche questo tipo di Diagramma è usato per complementare un diagramma statico, specialmente quando la logica implementata è troppo complessa per essere rappresentata efficacemente con un diagramma di sequenza.

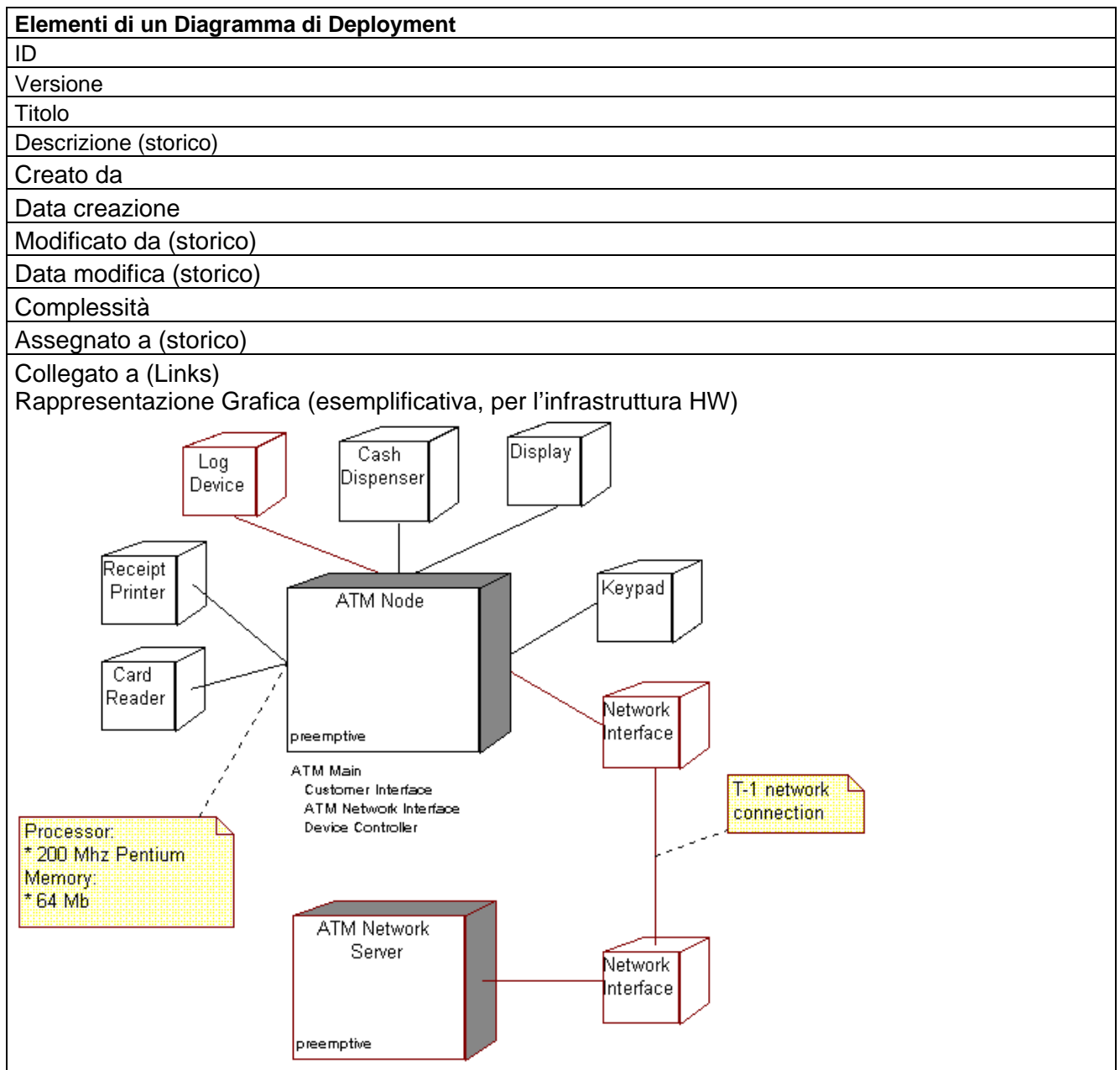
Il Diagramma di tipo Activity è di solito organizzato per colonne o righe, ognuna delle quali rappresenta quanto accade in uno dei componenti coinvolti nell'interazione.



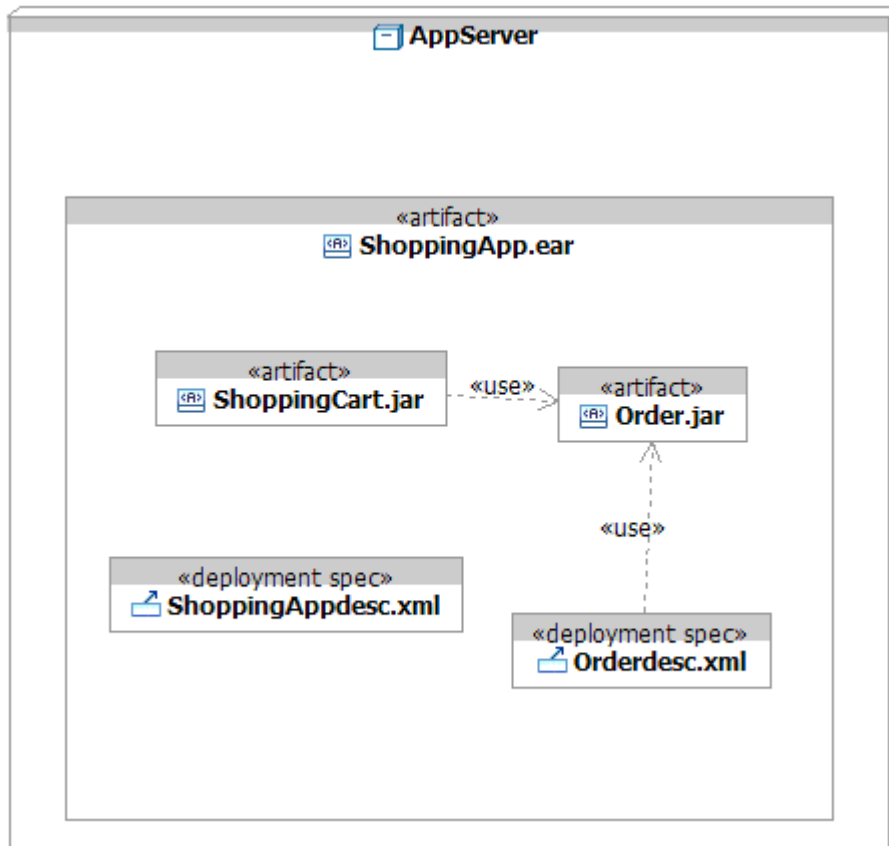
### 3.6 Diagramma di Deployment

Questo Diagramma schematizza a vario livello le componenti fisiche del sistema in sviluppo e le loro relazioni:

- Nodi (Sistemi HW)
- Sottosistemi (prodotti)
- Componenti (Moduli Applicativi)



Rappresentazione Grafica (esemplificativa, per le componenti applicative)





## 4 Processi Trasversali e tools di supporto

---

I processi su cui si concentra il presente documento sono quelli che possiamo identificare come processi che supportano l'intero Sistema Informativo come insieme dei progetti di sviluppo, e che definiscono attività di supporto o “trasversali” alla realizzazione del software.

Più in particolare:

- per quelli riferiti alla realizzazione (Software Implementation Processes)
  - I processi di gestione dei modelli di architettura applicativa e sistemistica
- per i processi di supporto allo sviluppo
  - la gestione della documentazione (sia interna che utente)
  - la gestione della configurazione del software inclusa la gestione dei problemi e delle modifiche
- per i processi relativi al riuso del software
  - gestione dei building block che costruiscono il sistema informatico, della loro evoluzione e la gestione del ciclo di vita degli asset applicativi

## 4.1 Processo di gestione della Configurazione

In questo processo è concentrata la gestione delle versioni del componente e delle workspaces di sviluppo, da mantenere in un repository centralizzato dal quale effettuare l'estrazione (check out) in caso di evoluzioni successive del componente stesso.

Devono essere gestiti dal processo di Gestione della Configurazione tutti i workproduct creati dalle fase precedenti che faranno parte della Baseline del prodotto/componente da rilasciare.

### **Obiettivi del Processo**

- Creare un repository di configurazione integrato contenente la baseline dei workproduct e dei componenti rilasciati dagli sviluppatori
- Fare la build dei componenti composti da più moduli sviluppati singolarmente
- Costruire gli eseguibili dei Componenti necessari al deployment negli ambienti di test

### **Risultati attesi**

- Creazione dei componenti complessi (Deployment Unit) da consegnare al gruppo di Test.
- Gestione delle versioni dei componenti
- Gestione delle versioni dei casi di test facenti parte della baseline del componente da rilasciare

### **Attori/Ruoli Aziendali coinvolti**

Sviluppatore: Figura tecnica specialista di sviluppo applicativo

Gestore della Configurazione Figura tecnica specialista di costruzione e gestione di ambienti di lavoro per la gestione della configurazione, produce la baseline e la build del componente complesso

### **Work products – input**

[CO] Codice delle Classi/Oggetti

[UT] Ambiente di Unit Test

[DU] Eseguibili per il deployment

[TF] Test Cases Funzionali

[TP] Test Plan Funzionale

[TI] Test Cases di Integrazione

[UC] Use Case Model

[IU] Specifiche interfacce utente

[ER] Modello concettuale dei dati

[AR] Diagramma Architeturale (Deployment)

[CL] Modello delle Classi e dei Componenti

[DB] Disegno fisico del DB

### **Output**

[DU] Eseguibili per il deployment

[BS] Baseline dei componenti da rilasciare e dei documenti/workproduct collegati

### **Attributi dell'oggetto di configurazione (Configuration Item)**

ID	Identificativo univoco (assegnato automaticamente)
Versione	Cambia ogni qualvolta un nuovo eseguibile è passato alla Fase di Test. (default: V0)
Titolo	
Descrizione	Breve descrizione delle caratteristiche degli elementi contenuti nella versione
Contenuto	Elenco degli elementi costituenti l'oggetto da sottoporre a configurazione (Casi di Test funzionali e di integrazione, Classi Software, Requisiti etc etc)
Stato	Rilasciato/in configurazione(baseline)
Fix collegate	Modifiche di soluzione di problemi contenute
Eseguibili	Lista degli eseguibili risultanti (la versione è la medesima del Componente)
Modelli Collegati	Vedi sotto

In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito
Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

### **Modelli usati nel processo**

	Componente
Diagramma di Deployment	Rm
Modello dei Componenti	Rm



- b. Output: eseguibili dei componenti in una specifica versione facenti riferimento alla baseline del sistema
- c. Attività:
  - i. Assemblaggio (build) dei componenti complessi
  - ii. Rilascio dei componenti in ambiente di test
  - iii. Raccolta dei log e dei risultati del test

#### 4. **Promozione o regressione della baseline:**

- a. Obiettivo: Questa fase del processo consiste nel promuovere la baseline del sistema e la versioni dei componenti verificati nella fase di test. Se il test restituisce risultati negativi, la baseline del sistema deve regredire allo stato di modifica e rimanere nel repository per essere aggiornata con le correzioni degli errori riscontrati
- b. Output: Baseline del sistema promossa all'ambiente di collaudo o regredita in modifica
- c. Attività:
  - i. Modifica dello stato della baseline del sistema
  - i. Passaggio della baseline promossa (intesa come insieme di tutti i workproduct che compongono quella versione del sistema) all'ambiente di collaudo se i test hanno dato esito positivo ovvero regredita allo stato di modifica se i test hanno dato esito negativo

## 4.2 Ingegneria dei Test

Il Ciclo di Vita del Test rappresenta l'intero processo di test richiesto per supportare il ciclo di sviluppo a partire dai requisiti funzionali per arrivare alla messa in produzione delle applicazioni.

Il ciclo include anche lo sviluppo e gestione degli ambienti di test e dei dati necessari allo svolgimento dei test.

Alcune tipologie di test (Test Statici o Ispezioni, Unit Test) sono eseguiti e controllati in autonomia dallo stesso sviluppatore e hanno di solito una gestione informale.

I test formali sono così categorizzati:

Tipologia	Descrizione
Functional test	Verifica che ad ogni fase di sviluppo ogni business function operi come stabilito nei requisiti e specificato nei documenti di design interno ed esterno.
Integration testing	Cerca di scoprire le eventuali collisioni tra diversi singoli moduli software integrati tra loro.
System testing	Verifica la corretta esecuzione dell'insieme dei componenti applicativi includendo interfacciamento verso altre applicazioni, considerando l'HW il SW ed una infrastruttura simile a quella di produzione.
Performance Testing	Misurare le prestazioni delle applicazioni e del sistema rispetto ai livelli di servizio attesi e definiti nei requisiti.
Regression testing	Cerca di scoprire le regressioni software, che potrebbero determinarsi quando alla variazione di alcuni moduli software alcune delle funzionalità che in precedenza funzionavano correttamente smettono di funzionare.
User Acceptance	Si tratta di un processo di test svolto da un esperto della materia e volto a verificare ed ottenere la conferma che una modifica o una richiesta aggiuntiva siano stati implementati correttamente rispetto a quanto definito nei requisiti e le funzionalità siano operative in modo corretto.
Operability Testing	Verifica che le applicazioni possano operare in maniera corretta sull'ambiente di produzione

Un dettaglio specifico merita il **test automatizzato ed in particolare il test di performance**, volto a misurare le prestazioni delle applicazioni e del sistema, in termini di tempi di risposta delle transazioni utente, della identificazione degli eventuali colli di bottiglia e fornendo informazioni sulla stabilità, scalabilità del sistema con l'obiettivo di stabilire in anticipo quello che sarà l'andamento del sistema di esercizio. I test partono da livelli di servizio attesi definiti nei requisiti e sono volti a verificare che le applicazioni rispettino tali livelli.

In particolare attraverso tool specifici si verifica con incrementi progressivi degli utenti virtuali quale siano i tempi di risposta della applicazione e quale sia il comportamento del sistema nell'ambiente dove sono in corso i test.

In particolare i tool di performance si avvalgono di alcune componenti essenziali per poter fornire informazioni utili al rispetto dei requisiti, ovvero:

- Strumenti di automazione del carico, dove sulla componente “master” risiede il repository dei test cases da eseguire e dove sulla componente “generatore di carico” viene simulato attraverso gli utenti virtuali il carico di sistema
- Strumenti di monitoraggio sistemi, che consentono di visualizzare e memorizzare durante la esecuzione dei test quale sia il comportamento del sistema in termini di rete, disco, memoria ed utilizzo dei processi
- Strumenti di Sonde Applicative, che monitorizzano i percorsi eseguiti dalle applicazioni, raccogliendo le informazioni sugli strumenti di diagnostica applicativa ed utili ad interpretare il comportamento della applicazione stessa

Poiché si vorrà simulare in anticipo su ambienti specifici di test quale sarà il possibile comportamento dell'ambiente di esercizio gli scenari di test prestazionali prevedranno anche la esecuzione contemporanea di più test per più applicazioni.

Un altro intento dei test di performance sarà quello di identificare il possibile punto di rottura con appositi scenari di carico di sistema, con lo scopo dunque di testare contemporaneamente sia i sistemi applicativi che la infrastruttura che ospita tali sistemi.

Il ciclo dei test di performance e' previsto nel master test plan e si uniforma a quanto evidenziato per gli altri livelli di test.

Anche per i test prestazionali esistono diverse tipologie di test che possono essere eseguiti e che devono essere stabilite nel master test plan, di seguito vengono identificati i più comuni

Tipologia	Descrizione
Performance Test	Determina e verifica la velocità o la capacità di risposta di un sistema
Test di Carico	Verifica il comportamento di una applicazione in condizioni normali ed in condizioni di picco
Stress Test	Verifica il comportamento di una applicazione quando questa viene sollecitata in condizioni normali ed in condizioni di picco
Test di durata	Verifica il comportamento di una applicazione quando al sistema viene richiesto di lavorare in condizioni di picco per un determinato intervallo di tempo
Test di Saturazione	Per verificare il comportamento di una applicazione quando il volume massimo di throughput (dati immessi)

Capacity test	Determina quanti utenti o transazioni un sistema e' in grado di reggere continuando a garantire i livelli di performance stabiliti/attesi
---------------	---

Un ulteriore approfondimento riguarda i test di Operabilità volti alla verifica che più applicazioni, all'interno di un sistema complesso fatto di interrelazioni tra le stesse applicazioni e le componenti degli altri sistemi software ed hardware, prima della loro messa in esercizio, possano operare in maniera corretta. Possono essere realizzati in contemporanea ai test di User Acceptance o temporalmente subito dopo come sequenza . Richiedono che non esistano difetti aperti e terminano quando il sistema sotto verifica rispetta i criteri stabiliti nel test plan.

La valenza di questa tipologia di test sta nel fatto che il superamento con esito positivo è in grado di assicurare che la organizzazione di supporto per l'ambiente di produzione sia preparata e possa realizzare il deploy degli oggetti, correlati al sistema sotto test, dagli ambienti di test, nonché sia in grado di poter supportare il sistema in oggetto una volta che ne e' stato fatto il deploy in produzione. Così facendo nell'ambiente di produzione si evitano difetti di tipo operativo .

### **Obiettivi del Processo**

Il processo è teso ad individuare i difetti inclusi nelle componenti fisiche del sistema, vale a dire le discrepanze tra il comportamento effettivo e quello atteso in base ai requisiti approvati.

### **Attori/Ruoli Aziendali coinvolti**

Sviluppatori: Figura tecnica specialista di sviluppo applicativo

Architetto del Test: progetta in collaborazione con il referente le specifiche dei test

Capo Progetto: contribuisce a stabilire la pianificazione dei test

Test Designer: realizza le specifiche tecniche dei test cases

Referente : Responsabile del Progetto o Area Applicativa

Test Executor: figura tecnica che esegue i test cases

Utenti (stakeholders): Rappresentanti delle linee operative che utilizzeranno il sistema ed eseguono i casi di test di accettazione del sistema

Sistemisti: Figura tecnica specialista di gestione dei sistemi IT , dei prodotti middleware e dei Database

### **Risultati attesi**

Tutti i Test Cases identificati nel Master Test Plan eseguono correttamente, senza evidenziare difetti.

### **Work products – input**

[PL] Principi, Linee Guida e Standards di riferimento



[PS] Performance SLA (service level agreement) metriche prestazionali che devono essere rispettate

[RQ] DB dei Requisiti

[UC] Use Case Model

[SQ] Sequence Diagram

[AD] Activity Diagram/Process Diagram

[IU] Specifiche interfacce utente

[AR] Diagramma Architeturale (Deployment)

[CL] Modello delle Classi e dei Componenti

[ER] Modello concettuale dei dati

[PR] WBS e Gantt del progetto

### **Output (modelli)**

[TS] Test Strategy

[TSC] Test Script

[TM] Master Test Plan

[IS] Static Test Plan

[TP] Detailed Test Plans (per level)

[TC] Test Cases

[TR] Test Report

[PD] Problem / Defect Log

### **Attributi dell'oggetto Test**

ID	Identificativo univoco (assegnato automaticamente)
Versione	Cambia se lo scopo (descrizione) del test è modificato su richiesta dell'utente. (default: V0)
Titolo	
Descrizione	Descrive lo scopo del test
Tipologia	
Stato	Planned/Failed/Passed
Complessità	Complessità di esecuzione (Alta/Media/Bassa)
Modalità di Esecuzione	Descrive i passi per effettuare l'esecuzione di un test manuale o punta allo script di un test automatizzato
Data di Esecuzione Prevista	Come da Master Test Plan
Data di Esecuzione Effettiva	Storico per ogni esecuzione (incluse le regressioni)
Pass/Fail	Storico per ogni esecuzione (incluse le regressioni)
Esecutore	Tester

Ambiente di esecuzione	Link al diagramma di deployment che descrive le specifiche dell'ambiente di esecuzione
Dati di esecuzione	Descrive i passi per effettuare la preparazione dei dati o punta allo script per il caricamento
Requisiti collegati	Requisiti che vengono verificati dal test
Modelli collegati	Vedi sotto

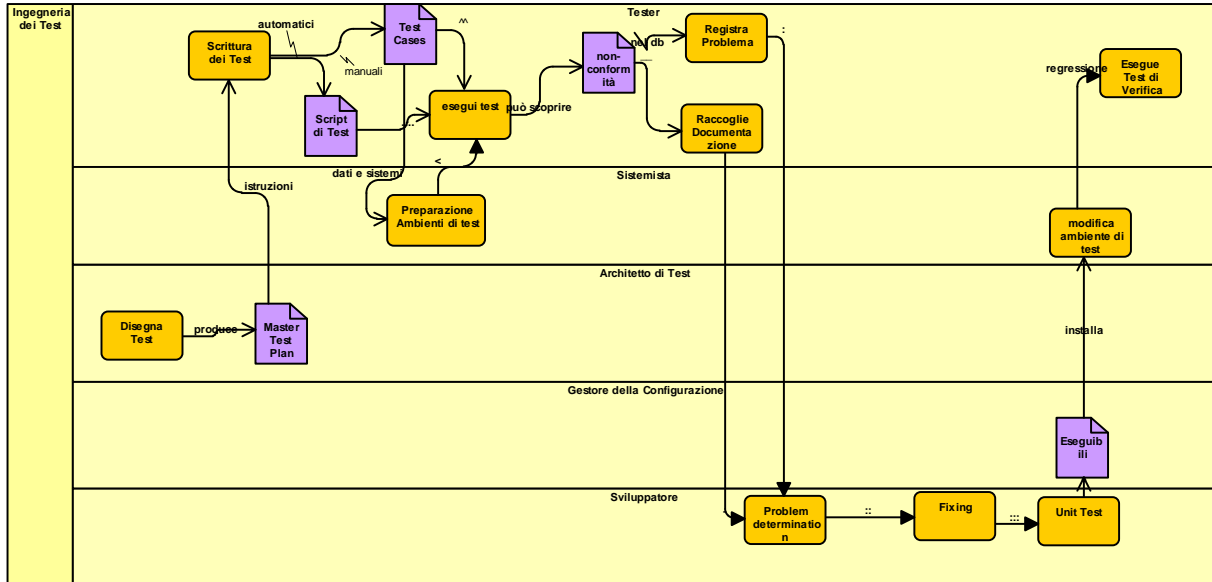
### **Modelli usati nella fase**

La definizione di dettaglio dei Test usa istanze di modelli UML analogamente alla fase di Disegno del Sistema e nella realizzazione degli Use Cases.

Link	Test Statico Unit Test	Functional Test, Regression Test,	User Acceptance	Integration Test, System Test	Performance Test	Operability Test
Modello dei Casi d'Uso	Ro	Wm	Wm	Wo		Wo
Diagramma di Sequenza	Ro	Wo	Wo	Wo	Wo	Wo
Modello delle Classi	m	O				
Diagramma di Deployment		Wm		Wm	Wm	Wm
Modello dei Componenti	Ro	Ro		Ro		
Activity Diagram	Ro	Wm		Wm	Wm	Wm
Process Diagram		Ro	Wm			Wm
Modello E/R		Ro		Ro		

## Attività

### Flusso generale del processo:



### 1. Pianificazione:

- Obiettivo:** questa fase include due momenti essenziali per poter definire correttamente quali saranno le fasi di test da realizzare.
- Output:** Master Test Plan.
- Attività:**
  - Definizione della strategia di test: un documento che rappresenta la linea guida ad alto livello di come si vuole affrontare il test e include la terminologia comune che identifica le criticità del sistema che sarà sottoposto a verifica.
  - Definizione del Master Test Plan: un documento che descrive nel dettaglio per ogni livello di test l'obiettivo, lo scopo, la tempistica ed il chi realizza' le attività' di verifica. Le tipologie i test del Master Test Plan sono strettamente correlate alla fase al ciclo di sviluppo del software.
  - Definizione della Tabella dei test (progettazione di alto livello dei test): i casi di test sono identificati sulla base della scomposizione funzionale e dei requisiti approntata in fase di preparazione e analisi dei requisiti, eventualmente con strumenti di gestione dei requisiti.

### 2. Design:

- Obiettivo:** La fase di design è volta a definire nel dettaglio gli elementi che costituiranno le verifiche da realizzare nel test.

- b. Output definizione di dettaglio dei casi di test (test cases), che contengono una serie di istruzioni, condizioni e variabili attraverso le quali è possibile determinare se un requisito e/o un caso d'uso (use case) per una applicazione sia parzialmente o totalmente soddisfatto.
- c. Attività:
  - i. Definizione di dettaglio dei test cases
  - ii. Predisposizione dei dati di test
  - iii. La gestione della configurazione dei test
  - iv. La predisposizione di un piano di esecuzione dei test
  - v. La predisposizione del o degli ambienti dove saranno eseguiti i test.

I test identificati in fase di pianificazione sono ripresi ed rianalizzati per determinare tutte le casistiche e condizioni di test da sottoporre a verifica, utilizzando a pieno le tecniche di progettazione del test (classi di equivalenza, tabella decisionale, valori limite, ecc). Il risultato di questa attività è lo **script di test**, realizzato secondo lo standard documentale previsto, con l'obiettivo di ottenere test facilmente riutilizzabili, riproducibili, oggettivi ed indipendenti, con la minor ridondanza possibile di informazioni, facilmente automatizzabili e mantenibili.

### 3. Esecuzione e Tracking:

- a. Obiettivo: questa fase vengono eseguiti i casi di test relativamente ad un dato sistema e si verifica che i risultati siano rispondenti ai risultati attesi. Le esecuzioni saranno reiterate e volte ad identificare eventuali difetti con lo scopo di poterli rimuovere per ottenere appunto i risultati attesi.
- b. Output : Test Results.
- d. Attività:
  - i. Esecuzione dei test inclusa la verifica della corrispondenza tra Documentazione e comportamento del sistema
  - ii. Monitoraggio con appositi tool per analizzare l'andamento delle applicazioni e dei sistemi sui vari ambienti (precedenti all'esercizio) durante la esecuzione dei test
  - iii. Registrazione dei Risultati
  - iv. Regression delle correzioni

### 4.3 Gestione dell'Architettura Applicativa e Tecnica

Questo processo assicura la consistenza, il riuso e l'ottimizzazione tecnica tra i vari progetti, verificando che gli stessi siano volti a costituire un insieme coerente di sistemi a supporto degli obiettivi dell'Istituto e allineati con gli standard adottati.

#### Obiettivi del Processo

- La progettazione di template architetturali per ogni tipologia di applicazione
- Definizione delle interfacce interne ed esterne dei componente software
- Tracciabilità della progettazione architetturale verso i requisiti software

#### Risultati attesi

- Mantenimento di una visione architetturale complessiva (applicativa e sistemistica) .
- Approvazione e diffusione degli standard tecnici
- Revisione ed approvazione dell'architettura e delle scelte tecniche dei singoli progetti

#### Attori/Ruoli Aziendali coinvolti

Architetto Applicativo: Figura tecnica specialista di sviluppo applicativo

Architetto di Infrastruttura: Figura tecnica specialista di costruzione e gestione di sistemi infrastrutturali hardware e middleware

#### Work products – input

[ST] Specifiche Tecniche del singolo Progetto

[PL] Principi, Linee Guida e Standards di riferimento

#### Output

[AA] Architettura Applicativa dell'Istituto

[AB] Architettura dei processi dell'Istituto

[AI] Architettura tecnica dell'Istituto

[AD] Architettura dati ed informazioni dell'Istituto

#### Attributi dell'oggetto di Architettura

ID	Identificativo univoco (assegnato automaticamente)
Versione	(default: V0)
Titolo	
Descrizione	Breve descrizione dello scopo dell'Architettura
Contenuto	Elenco degli elementi costituenti l'architettura (Diagrammi di Deployment, Casi d'Uso, Classi Software,Requisiti etc etc)
Modelli Collegati	Vedi sotto

In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

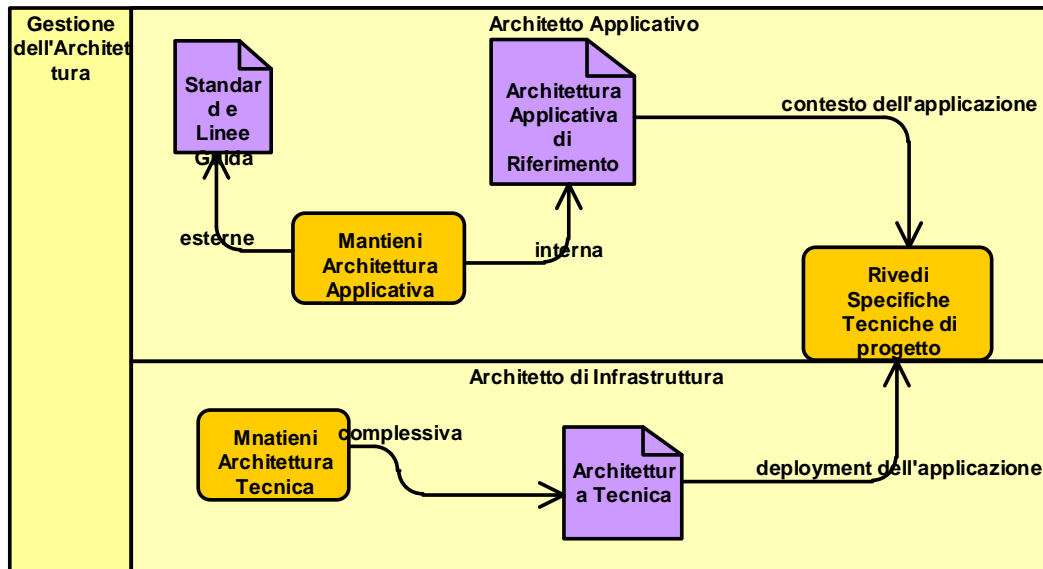
Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito
Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

### **Modelli usati nel processo**

	Architettura
Diagramma di Deployment	Wm
Modello delle Classi	Wm
Modello E/R	Wm
Sequence Diagrams	Wm
Modello dei Componenti	Wm
Activity Diagram	Wm
Process Diagrams	Wm
Casi D'Uso	Wm

## Attività

### Flusso generale del processo



#### 1. Creazione Modelli di Istituto

- a. Obiettivo: Questa fase mira a creare i modelli di contesto in cui si inseriscono i vari progetti di sviluppo applicativo e di infrastruttura
- b. Output : Architetture di riferimento
- c. Attività:
  - i. Ricognizione architetture esistenti
  - ii. Raccolta Standards e Policies
  - iii. Scrittura Documenti di Architettura
  - iv. Manutenzione delle Architetture in relazione a nuove scelte e decisioni

#### 2. Revisioni Architetture

- d. Obiettivo: Verificare che l'architettura tecnica del singolo progetto sia conforme con le Architetture di riferimento
- e. Output: Approvazione delle Specifiche Tecniche del progetto
- f. Attività:
  - i. Incontro con gli architetti di progetto
  - ii. Presentazione delle raccomandazioni al progetto
  - iii. Aggiornamento delle Architetture di Istituto.

## 4.4 Processo di Gestione della Documentazione

In questo processo è concentrata la gestione della documentazione prodotta durante le varie fasi dello sviluppo.

Per documentazione si intende tutto ciò che viene prodotto per descrivere le varie parti del sistema (principalmente i work-products creati dalle varie fasi del processo), i documenti di supporto all'installazione del sistema, alla sua gestione in esercizio e ai documenti di supporto all'uso del sistema (manuali utente).

### **Obiettivi del Processo**

- Creare un repository centralizzato di gestione della documentazione (baseline della documentazione)
- Creare una documentazione erogabile come corso di formazione per l'utilizzo del sistema da rilasciare
- Creare il "BOM" (Bill of Material) che specifica tutti i componenti e tutti i documenti necessari alla gestione del sistema
- Creare le note sul rilascio (release notes) che descrivono le caratteristiche principali del prodotto

### **Risultati attesi**

- Creazione del materiale tecnico per la gestione del sistema
- Creazione del materiale didattico per l'addestramento degli utenti
- Realizzazione del software per la fruizione dei contenuti da utilizzare in modalità on-line

### **Uscita dalla Fase**

1. Il materiale di descrizione del sistema è stato memorizzato nel repository centralizzato
2. Il materiale di utilizzo del sistema è stato creato e memorizzato nel repository centralizzato
3. Il materiale per l'addestramento degli utenti è stato realizzato e memorizzato nel repository centralizzato

### **Attori/Ruoli Aziendali coinvolti**

**Redattore Tecnico:** (Specializzazione di Disegnatore Applicativo) Figura tecnica specialista nello sviluppo di documentazione di supporto per l'utente finale

**Formatore:** (Specializzazione di Disegnatore Applicativo) Figura tecnica specialista nello sviluppo di documentazione di formazione utente.



### **Work products – input**

- [CO] Codice delle Classi/Oggetti
- [UT] Ambiente di Unit Test
- [DU] Eseguibili per il deployment
- [TF] Test Cases Funzionali
- [TP] Test Plan Funzionale
- [TI] Test Cases di Integrazione
- [UC] Use Case Model
- [IU] Specifiche interfacce utente
- [ER] Modello concettuale dei dati
- [AR] Diagramma Architeturale (Deployment)
- [CL] Modello delle Classi e dei Componenti
- [DB] Disegno fisico del DB

### **Output**

- [DO] Repository centralizzato contenente la documentazione del sistema
- [BS] Baseline dei documenti di supporto congruente con la versione del sistema rilasciato

### **Attributi dell'oggetto "Documento"**

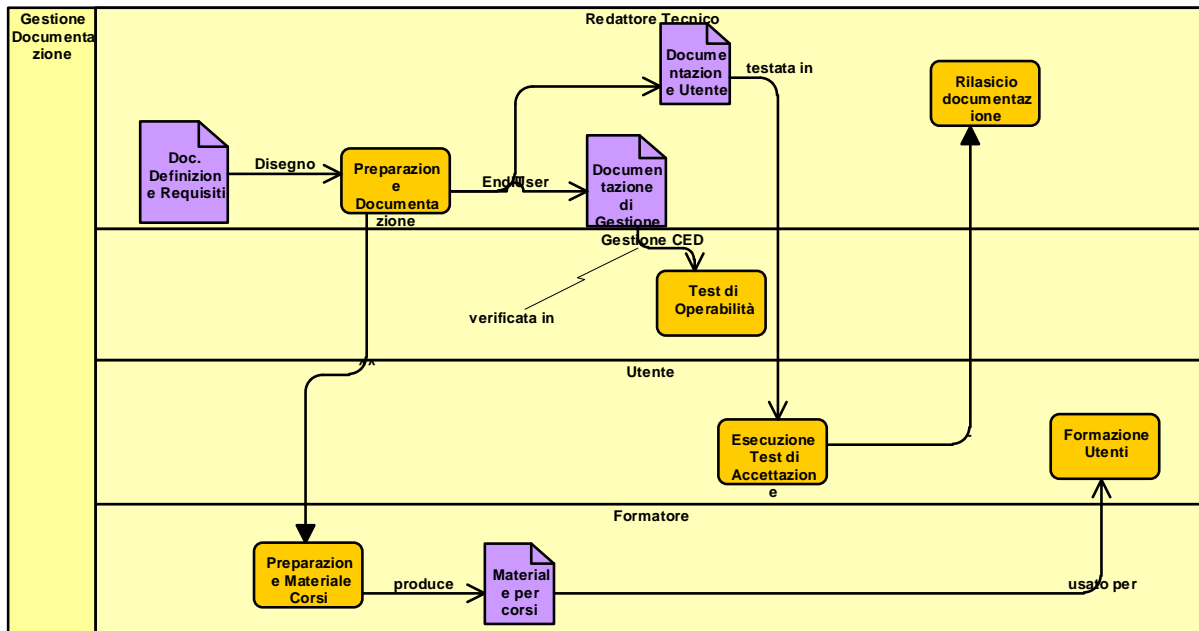
ID	Identificativo univoco (assegnato dal tool di gestione documentale)
Versione	Cambia ogni qualvolta un documento è aggiornato (default: V0)
Titolo	Titolo documento
Descrizione	Descrizione del Documento
Stato	In Lavorazione/Rilasciato
Fix collegate	Modifiche al contenuto del documento
Modelli Contenuti nel documento	Lista dei modelli contenuti nel documento

In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito
Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

## Attività

### Flusso generale del processo



#### 1. Creazione Repository di Documentazione Centralizzato

- Obiettivo:** Questa attività crea il repository centralizzato che dovrà contenere i documenti prodotti nelle fasi primarie del ciclo di vita del software e i documenti a supporto dell'uso e della gestione in esercizio del sistema
- Output:** Repository dei documenti
- Attività:**
  - Creazione del repository

#### 2. Creazione della documentazione di gestione

- Obiettivo:** Creare la documentazione che consenta di installare e gestire il sistema nei diversi ambienti.
- Output:** Documenti tecnici d'installazione e gestione del sistema
- Attività:**
  - Creazione della documentazione tecnica di gestione
  - Verifica (test) della documentazione

#### 3. Creazione della documentazione utente:

- Obiettivo:** Creazione della documentazione utente per l'utilizzo del sistema e della documentazione per l'addestramento degli utenti.

- b. Output: Manuali utente e materiale per la fruizione del contenuto on-line. Materiale didattico per l'addestramento degli utenti erogabile come corso di formazione.
  - c. Attività:
    - i. Creazione dei manuali utente
    - ii. Creazione del software per la fruizione on-line dei contenuti utente
    - iii. Creazione del materiale per il corso di formazione utente
    - iv. Verifica del materiale creato
4. **Rilascio della documentazione agli utenti:**
- a. Obiettivo: Creazione della baseline della documentazione del sistema.
  - b. Output: Baseline della documentazione da rilasciare agli utenti
  - c. Attività:
    - i. Creazione della baseline dei documenti allineata alla versione del sistema rilasciato
    - ii. Rilascio dei documenti agli utenti

## 4.5 Processo di Riuso (Gestione Asset Riutilizzabili)

In questo processo è definita la modalità di gestione degli assets riutilizzabili, al fine di condividere per riutilizzare componenti e documenti di progettazione prodotti in altri progetti o settori aziendali.

Per Asset riutilizzabile si intende tutto ciò che può essere messo a disposizione di altri progettisti, sviluppatori, tester, utenti e stakeholders in genere, per essere riutilizzato.

Gli assets riutilizzabili devono avere una loro specifica descrizione e devono essere catalogati su un opportuno strumento di condivisione.

### **Obiettivi del Processo**

- Creare un repository centralizzato di gestione degli assets riutilizzabili
- Catalogare gli assets riutilizzabili in modo che siano facilmente reperibili e scaricabili dal repository
- Mantenere traccia degli utilizzi al fine di comunicare agli utenti modifiche agli assets che stanno usando

### **Risultati attesi**

- Creazione del repository per la gestione degli assets
- Creazione di reports statistici sull'utilizzo degli assets catalogati

### **Uscita dalla Fase**

1. Il repository degli assets è stato creato
2. Sono stati definiti i criteri di utilizzo degli assets

### **Attori/Ruoli Aziendali coinvolti**

Gestore Assets: Figura tecnica preposta alla gestione degli assets nel repository

Sviluppatore: Figura tecnica specialista nello sviluppo di specifiche, componenti software, documentazione tecnica e utente

### **Work products – input**

[CO] Codice delle Classi/Oggetti

[DU] Eseguibili per il deployment

[TF] Test Cases del componente riutilizzabile

[UC] Use Case Model del componente riutilizzabile

[IU] Specifiche interfacce utente del componente riutilizzabile

[ER] Modello concettuale dei dati

[AR] Documenti su Architetture di riferimento

[CL] Modello delle Classi e dei Componenti

## **Output**

[RS] Repository centralizzato contenente gli assets riutilizzabili

### **Attributi dell'oggetto "Asset"**

ID	Identificativo univoco (assegnato dal tool di gestione documentale)
Versione	Cambia ogni qualvolta un asset è aggiornato
Titolo	
Descrizione	Descrizione dell'asset
Stato	Approvato/Rilasciato "as is"
Fix collegate	Modifiche richieste all'asset
Lista componenti asset	Lista dei componenti software o documentali dell'asset
Specifiche di utilizzo	Particolari indicazioni sull'utilizzo (ambito di applicazione, problema risolto dall'asset etc etc)
Asset relazionati	Relazioni con altri asset che sono necessari per corretto funzionamento dell'asset che si sta valutando
Policy di utilizzo	Regole di utilizzo dell'asset (sicurezza, copyright, distribuzione, modifica della struttura o evoluzione tecnologica)
Modelli Contenuti nell'asset	Lista dei modelli contenuti nell'asset

In quanto si vuole tenere traccia di tutte le modifiche, per ognuna di esse vanno valorizzati gli attributi di seguito specificati (cosiddetti attributi di variazione).

Data Modifica	Data della modifica degli attributi del requisito che danno luogo a nuova versione
Richiedente Modifica	Nominativo di chi ha richiesto la modifica
Autore Modifica	Nominativo di chi ha modificato il requisito
Motivo Modifica	Descrizione del motivo della richiesta di modifica.
Valorizzazione	Costo aggiuntivo della modifica del requisito.

## **Attività**

### **1. Creazione Repository di Gestione Assets Centralizzato**

- a. **Obiettivo:** Questa attività crea il repository centralizzato che dovrà contenere gli assets riutilizzabili
- b. **Output:** Repository degli assets riutilizzabili
- c. **Attività:**
  - i. Creazione del repository

### **2. Raccolta e Catalogazione degli Assets riutilizzabili:**

- a. **Obiettivo:** Raccogliere e catalogare asset riutilizzabili prodotti dai diversi gruppi di lavoro progettuali



- b. Output: Asset catalogati nel repository
- c. Attività:
  - i. Valutazione asset esistenti per verificarne l'effettiva possibilità di riutilizzo
  - ii. Catalogazione degli assets nel repository

**ALTRI DOCUMENTI DEL CICLO DI VITA DEL SOFTWARE IN INPDAP**

<b>DOCUMENTI RELATIVI ALLA PRODUZIONE DEL SOFTWARE</b>		
<b>Codice</b>	<b>Titolo</b>	<b>Abstract</b>
A0-C03-0200-001	Standard del documento di Descrizione Processo	Descrive la struttura e le modalità di compilazione del documento "Descrizione del processo", in cui si descrivono il processo relativo al trattamento di una pratica istituzionale, con riferimento a quanto rappresentato nello schema IDEF0 relativo (che deve essere prodotto prima della redazione del documento), nonché eventuali requisiti funzionali e non emersi negli incontri finalizzati alla definizione del processo. Il documento costituisce uno dei prodotti dell'attività di consulenza di progetto.
A0-C03-0200-002	Standard del documento Specificazione del Processo Informatico	Illustra la struttura e le modalità di compilazione del documento "Specificazione del Processo Informatico", in cui si descrive la soluzione tecnico/funzionale che si propone all'Utente per il raggiungimento degli obiettivi espressi nel documento "Descrizione del Processo".
A0-C04-0500-001	Descrizione del Ciclo di Vita del software	Descrive il ciclo di vita del software da adottare nell'ambito dello sviluppo del Sistema Informativo Normalizzato dell'INPDAP.
A0-C04-0500-003	Descrizione del Ciclo di Vita Ridotto del Software	Descrive il ciclo di vita ridotto del software da adottare: <ul style="list-style-type: none"><li>• per progetti di piccole dimensioni o di ridotta complessità;</li><li>• per progetti con durata limitata in cui risulterebbe difficile e poco opportuno seguire e rispettare i tempi e le fasi previsti nel ciclo di vita classico;</li><li>• per progetti che sviluppano componenti di servizio generalizzate non equiparabili ad applicazioni gestionali fruibili direttamente dall'utente.</li></ul>
A0-C04-0500-004	Descrizione del Ciclo di Sviluppo del Portale	Descrive il ciclo di vita del software da adottare per progettare, sviluppare e rilasciare portali.

<b>DOCUMENTI RELATIVI ALLA PRODUZIONE DEL SOFTWARE</b>		
<b>Codice</b>	<b>Titolo</b>	<b>Abstract</b>
A0-C04-0500-005	Descrizione del Ciclo di Produzione Servizi Software	Descrive il ciclo di vita del software da adottare per progettare, sviluppare e rilasciare un servizio od un metodo fornito all'esterno dei confini dell'applicazione.
A0-C04-0500-006	Descrizione del Ciclo per gli interventi di MEV	Descrive il ciclo di vita ridotto del software da adottare per gli interventi di manutenzione evolutiva.
A0-C03-0500-002	Standard del documento di Specifica dei Requisiti	Definisce la struttura, i contenuti e le modalità di compilazione del documento di "Specifica dei Requisiti", in cui si descrivono le esigenze dell'utente e si individua il modo più opportuno per soddisfarle.
A0-C03-0500-003	Standard del documento di Specifica di Progettazione	Definisce la struttura, i contenuti e le modalità di compilazione del documento di "Specifica di Progettazione" che costituisce il documento a valenza contrattuale, prodotto al termine della fase di Progettazione di un obiettivo sviluppato secondo la metodologia object oriented.
A0-C03-0500-011	Standard del documento di Specifica di Progettazione Semplificato	Definisce la struttura, i contenuti e le modalità di compilazione del documento di "Specifica di Progettazione" che costituisce il documento a valenza contrattuale, prodotto al termine della fase di Progettazione di un obiettivo sviluppato secondo la metodologia object oriented. Da utilizzare nel caso si adotti un ciclo di vita semplificato.
A0-C03-0500-014	Standard del documento di Specifica di Progettazione per gli interventi di MEV	Definisce la struttura, i contenuti e le modalità di compilazione del documento di "Specifica di Progettazione" per gli interventi di MEV.



DOCUMENTI RELATIVI ALLA PRODUZIONE DEL SOFTWARE		
Codice	Titolo	Abstract
A0-C03-0500-012	Standard del documento di Specifica di Integrazione	Definisce la struttura, i contenuti e le modalità di compilazione del documento di "Specifica di Integrazione", il cui scopo è descrivere le modalità con cui gli applicativi verticali possono utilizzare le funzionalità rese disponibili da un modulo trasversale comune indicando: <ul style="list-style-type: none"><li>• le modalità operative con cui l'applicativo verticale deve interfacciare con il modulo;</li><li>• l'eventuale impatto sulla configurazione dell'applicazione che integra il modulo.</li></ul>
A0-C03-0500-013	Standard di realizzazione procedure batch	Descrive l'architettura applicativa per lo sviluppo delle applicazioni batch. In particolare, vengono illustrate le componenti software che compongono tale architettura, le regole per la corretta realizzazione del software e le norme di utilizzo di componenti centralizzati realizzati allo scopo di fornire strumenti standardizzati in grado di risolvere problemi comuni a tutte le procedure.
A0-C03-0700-003	Standard del Manuale di gestione del batch	Descrive le modalità per la redazione del "Manuale di gestione del batch" di un'area applicativa, nel quale devono essere descritte in maniera completa le caratteristiche generali <ul style="list-style-type: none"><li>• dell'applicazione di cui le procedure batch oggetto del manuale fanno parte in termini di caratteristiche funzionali e di ambiente elaborativo;</li><li>• dei batch dell'area applicativa, in termini di ambiente elaborativo, di modalità di schedulazione, di modalità di controllo dell'esito dello stesso e di gestione degli errori).</li></ul>
A0-C03-0500-005	Standard del Manuale utente	Descrive le modalità di compilazione del documento "Documentazione e Manualistica utente".

DOCUMENTI RELATIVI ALLA PRODUZIONE DEL SOFTWARE		
Codice	Titolo	Abstract
A0-C03-0500-006	Standard di Interfaccia di presentazione delle applicazioni Web	Esponde le linee guida per la progettazione delle interfacce di presentazione delle applicazioni Web. In particolare pone attenzione su quegli aspetti che possono garantire l'efficacia nell'uso, la navigabilità delle funzioni, l'omogeneità del layout, la coerenza dell'interazione, la facilità di utilizzo da parte dell'utente, in una parola l'usabilità delle procedure Web realizzate nell'ambito del Programma.
A0-C03-0500-008	Standard di realizzazione e nomenclatura oggetti software	Indica le linee guida, le best practices e le metodologie di sviluppo per le applicazioni J2EE. In particolare il documento illustra le componenti software che compongono tale architettura e le regole per la corretta realizzazione del software, sia in ambiente web che batch.
A0-C03-0500-007	Standard dei Documenti di Test (Piano di test, progettazione ed esecuzione)	Illustra il processo da seguire nell'ambito di un obiettivo di sviluppo per lo svolgimento dell'attività di test di sistema ed i prodotti da realizzare nelle diverse fasi del processo stesso. Descrive inoltre lo standard relativo ai seguenti documenti previsti dal CVS:  "Tabella dei test (Progettazione Condizioni e casi di test di alto livello)"  "Specifiche di test (Progettazione di dettaglio – script di test)"  "Esecuzione dei casi di Test"
A0-C02-0100-004	Linee guida per la gestione della configurazione del software	Fornisce le linee guida per governare la configurazione delle applicazioni software e multimediali rilasciate all'utente. Definisce le modalità ed i criteri per registrare gli interventi di variazione alla configurazione e gestirne l'archiviazione.

<b>DOCUMENTI RELATIVI ALLA PRODUZIONE DEL SOFTWARE</b>		
<b>Codice</b>	<b>Titolo</b>	<b>Abstract</b>
A0-C03-0100-011	Standard del Piano di gestione della configurazione	Fornisce le regole da seguire per la redazione del Piano di gestione della configurazione, in contesti hardware, software e documentali.
A0-C03-0900-001	Modellazione dati	Descrive il processo di progettazione di una base dati relazionale, secondo una strategia che prevede, a partire dall'analisi dei requisiti, la progettazione di uno schema concettuale, le fasi di semplificazione e traduzione che portano alla costituzione dello schema logico, l'uso delle tecniche di normalizzazione come verifica degli schemi stessi. Il documento ha lo scopo di giungere ad un linguaggio comune, condividendo le definizioni di base e indicando i formalismi da utilizzare per la rappresentazione dei dati secondo la rappresentazione UML nell'ambito del prodotto Enterprise Architect scelto come strumento per la definizione e la documentazione dei dati.
A0-C03-0900-006	Standard del Documento di Analisi Banca Dati	Indica quali sono i contenuti del documento che verrà fornito all'utente per la comprensione e la documentazione dello schema dati, a livello concettuale.
A0-C03-0900-002	Standard del Documento di Progettazione Banca Dati	Illustra le modalità di compilazione di ognuno dei paragrafi che compongono l'indice del documento "Documento di Progettazione Banca Dati" il cui scopo è quello di contenere le informazioni relative ai dati di interesse per l'istituto e le relazioni tra di essi, rappresentate secondo il modello relazionale dei dati.
A0-C03-0900-003	Standard del Documento di Realizzazione Banca Dati	Fornisce lo standard per la realizzazione del documento "Realizzazione banca dati" il cui scopo è quello di fornire le informazioni necessarie al gruppo DBA per implementare lo schema fisico dei dati.

<b>DOCUMENTI RELATIVI ALLA PRODUZIONE DEL SOFTWARE</b>		
<b>Codice</b>	<b>Titolo</b>	<b>Abstract</b>
A0-C03-0900-007	Standard di nomenclatura Banca Dati	Fornisce lo standard di nomenclatura per la definizione degli schemi concettuali, logici e fisici di una base dati.
A0-C03-0900-008	Standard del documento di Migrazione banca dati	Fornisce lo standard per la redazione del documento da produrre per la migrazione dati da una struttura preesistente ad una nuova banca dati.
A0-C03-0700-001	Definizione architettura tecnica di riferimento	Definisce l'architettura complessiva del nuovo sistema informativo Istituzionale Inpdap, individuando e descrivendo i diversi componenti che la costituiscono: <ul style="list-style-type: none"><li>• sottosistema di accesso</li><li>• sottosistema applicativo</li><li>• sottosistema di interscambio</li><li>• servizi trasversali</li><li>• architettura tecnologica</li><li>• sicurezza.</li></ul>
A0-C03-0900-009	Standard di gestione delle Tabelle Tipologiche	Stabilisce le modalità operative per la documentazione delle classi di dati cosiddette "Tipologiche" o di "Validazione" allo scopo di facilitarne la condivisione ed evitare duplicazioni
A0-C03-0500-010	Specifiche degli standard di produzione CBT	Fornisce linee guida agli sviluppatori di un courseware al fine di creare omogeneità funzionale tra i diversi courseware e azzerare l'impatto per l'utente nell'approccio ad un nuovo courseware attraverso la conformità operativa e topografica degli strumenti funzionali e di navigazione. Il documento presenta, a titolo di esempio, schermate relative ai diversi ambienti costruite secondo i criteri generali
A0-C03-0500-009	Standard di documentazione e manualistica per la struttura di esercizio	Fornisce lo standard del documento da redigere per il rilascio del software alla struttura di gestione esercizio

<b>DOCUMENTI RELATIVI ALLA PRODUZIONE DEL SOFTWARE</b>		
<b>Codice</b>	<b>Titolo</b>	<b>Abstract</b>
A0-A04-1700-005	Linee Guida per l'integrazione delle applicazioni web con il sistema di Access Management	Definisce le modalità di integrazione delle applicazioni da sviluppare con il sistema di Access e Identity Management in uso presso l'Istituto.
A0-A04-1400-002	Linee guida sull'utilizzo dei componenti trasversali infrastrutturali	<p>Il documento ha lo scopo di fornire le informazioni e i criteri generali sull'utilizzo dei componenti trasversali infrastrutturali , per ciascuno dei componenti vengono indicati:</p> <ul style="list-style-type: none"><li>• la descrizione del componente,</li><li>• i criteri per il suo utilizzo mediante l'integrazione negli applicativi,</li><li>• gli aspetti organizzativi che possono avere impatto sui progetti che li utilizzano.</li></ul>