ALIM2

Offerta per la realizzazione della biblioteca digitale di testi mediolatini ALIM2

1. Executive Summary

Nell'elaborazione della proposta per la realizzazione del prototipo del sistema ALIM2, Net7 srl si è posta in primo luogo l'obiettivo di rispondere ai vincoli tecnici irrinunciabili enucleati nelle specifiche richieste.

L'implementazione della biblioteca digitale ALIM2 sarà supportata da software di qualità, sviluppato a partire da componenti di base affidabili, ricchi di funzionalità, prestigiosi in termini di diffusione nel mercato e pienamente compatibili con gli standard tecnologici più diffusi del mondo web. Tutti i prodotti software di base scelti (database, server http, triplestore etc) sono rilasciati con licenza Open Source e basati su standard aperti, cosa che garantisce:

- un rischio nullo di lock-in legato all'uso di formati e software proprietari;
- minore rischio di obsolescenza tecnologica;
- maggiore sicurezza.

Il team di lavoro scelto da Net7 è composto da professionisti di grande esperienza nel settore delle Digital Humanities.

La realizzazione del software sarà poi condotta nel rispetto di standard metodologici di grande affidabilità. Nel progetto sarà posta estrema cura alle metodologie di lavoro, che seguiranno standard di qualità conformi alla norma ISO, tecniche di project management ispirate ai principi del Project Management Institute e l'uso di strumenti moderni ed efficaci per supportare i processi di sviluppo e più genericamente la conduzione del progetto.

La proprietà del software prodotto ad hoc per il progetto sarà del committente, a cui verranno rilasciati i codici sorgenti senza alcun vincolo né relativo al suo uso, né relativo alla sua distribuzione, né di qualunque altro genere.

2. La proposta per il sistema ALIM2

2.1 Uno sguardo d'insieme

La presente proposta risponde alla richiesta di realizzare la piattaforma ALIM2 - biblioteca digitale di testi mediolatini, che affianchi e sostituisca l'attuale sito: http :// alim . dfll . univr . it /.

Il sistema proposto implementa un sistema minimale di requisiti tecnici, descritti in dettaglio di seguito:

- il caricamento e la gestione dei testi XML-TEI
- l'esportazione degli XML-TEI in formati standard (TXT, PDF, HTML)
- il caricamento e la gestione dell'ontologia di riferimento
- la gestione di pagine e contenuti della Digital Library

- la ricerca semantica.

Basato sin dall'inizio su software open source e sugli standard del Semantic Web e Linked Data, ALIM2 è progettato per essere un sistema complesso, che, in una fase più avanzata del progetto, può rendere possibile:

- l'annotazione semantica dei testi e l'accesso ai frammenti di testo annotati e alle annotazioni tramite interfacce grafiche e meccanismi di ricerca adeguati;
- il popolamento di vocabolari e l'inserimento di asserzioni sulle istanze attraverso interfacce grafiche e meccanismi di ricerca adeguati;
- l'accesso ai vocabolari e la loro gestione;
- l'integrazione con Linked Data provider come Europeana, DBpedia, etc.

2.2 L'architettura generale

ALIM 2 consiste in un'applicazione web basata su un'architettura multi-layer, con una divisione tra gli strati di gestione dei dati (*data layer*), logica applicativa (*business layer*) e di interfacciamento verso l'esterno (*front-end layer*), indicando con quest'ultimo punto sia gli aspetti di presentazione verso gli utenti finali che l'integrazione applicativa con altri sistemi software.

Nel rispetto del principio della *separazione delle responsabilità*, gli strati applicativi del sistema saranno realizzati in ottica modulare: la loro integrazione si baserà principalmente su uno strato di interfacce applicative (API) formalmente definite e implementate tramite web services di tipo RESTful.

Lo "strato dati", come mostrato nel diagramma, è piuttosto articolato, essendo basato su più sistemi, un database relazionale, un triple store e, nella versione finale, un text search engine, ognuno specializzato ad assolvere specifiche funzioni applicative.

Il database, implementato con la componente MariaDB (pienamente compatibile con MySQL), contiene:

- le informazioni di "profilazione" degli utenti e una descrizione dei loro ruoli.
- i testi da analizzare. Al caricamento dei file XML/TEI, uno script interpreterà i metadati contenuti nell'header TEI convertendoli in triple, sulla base della mappatura con le ontologie OWL. Tali triple saranno riportate opportunamente nel triplestore, mentre nel database saranno memorizzati i testi, in multiple versioni (ad esempio la versione annotabile, la versione fruibile dal front-end).

I dati RDF, annotazioni e metadati dei testi, sono archiviati in un triplestore, Sesame, compatibile con il protocollo SPARQL e che offre pieno supporto per i named graphs.

L'indicizzazione "full-text" dei metadati e delle risorse avverrà tramite il motore di ricerca open source Solr.

La componente principale del Business Layer è rappresentata dalle *Annotation Server API*: consistono in web services REST che astraggono completamente logica di

gestione delle annotazione, la scrittura dei dati, il loro reperimento nonché l'organizzazione delle annotazioni in *notebook*. Il sistema si presenta così naturalmente aperto: è possibile creare diversi tipi di *client* per fornire e ricevere dati dal sistema.

Sempre afferenti allo strato di logica intermedio sono le cosiddette *Utilities*, un insieme di applicazioni che in ottica batch permettono di eseguire operazioni sui dati del sistema. Come Utilities saranno implementati:

- un tool che, in parallelo alla possibilità di caricare file XML/TEI via web, consentirà l'inserimento "massivo" di risorse testuali (*Batch TEI Upload*)
- il modulo per il caricamento delle ontologie (*Gestione delle ontologie di dominio*)

Per quanto riguarda lo strato di *Front-End*, distinguiamo qui le interfacce web della Digital Library, un sito per la gestione delle informazioni sul progetto destinate agli operatori (sia del front che del back-office), e le API applicative che facilitano l'accesso delle informazioni gestite in ALIM 2 ad agenti software. Basilare in tutte e tre le implementazioni è un modulo per sovraintendere ogni aspetto di gestione della sicurezza e della privacy, dalla profilazione e autenticazione degli utenti alla gestione dei loro ruoli fino alla definizione di policy precise per l'accesso ai dati.

Nel sistema web della Digital Library l'utente accederà alle interfacce di amministrazione del back-office e avrà a disposizione un ambiente di lavoro per gestire i diversi moduli della biblioteca digitale che rispondono alle sue specifiche esigenze. Qui l'utente può caricare file OWL e file XML-TEI. L'interfaccia del front-office offre poi all'utente generico la possibilità di effettuare ricerche, di sfogliare le singole risorse presenti nella biblioteca digitale e di visualizzarne i metadati.

Il sito ALIM 2, collegato alla biblioteca digitale, presenta le informazioni relative al progetto e ogni contenuto altro dalla biblioteca digitale vera e propria e sarà di facile gestione, basato su software open source (Wordpress).

Infine l'accesso ai dati di ALIM 2 sarà possibile anche via interfacce applicative, a partire dalla possibilità di interagire, tramite meccanismi di *access control*, con lo SPARQL Endpoint esposto dal Triple Store.

2.3 Descrizione del software di base

ALIM2 consiste in un'applicazione web, le cui componenti "server side" sono sviluppate in linguaggio PHP, per quanto riguarda gli aspetti di gestione dell'interfaccia, e JAVA per la comunicazione applicativa con il Data Layer.

Il sistema sfrutta delle componenti applicative di base per l'implementazione dei suoi servizi, tutte basate su software open source: tralasciando i moduli "standard" delle realizzazioni Intranet (sito web Apache con interprete PHP mod_php, servlet engine Java Apache Tomcat) vale la pena di spendere qualche parola per descrivere i software scelti per l'implementazione del Data Layer, ovvero il DBMS MariaDB, un fork di MySQL completamente compatibile con quest'ultimo, il triple store Sesame e il Search Engine Solr.

MariaDB

MariaDB (http://mariadb.org/) si propone come un "drop-in replacement" del celeberrimo DBMS MySQL, indubbiamente il database open source più diffuso al mondo. Si tratta infatti di un progetto fondato dallo stesso creatore di MySQL, Michael "Monty" Widenius, per poter garantire il massimo controllo e licenze più permissive dopo che la sua creatura MySQL era passata sotto il controllo della corporation Oracle.

MariaDB è quindi completamente compatibile con MySQL con cui condivide buona parte del design architetturale. Le sue caratteristiche lo rendono estremamente veloce e facile da integrare con tutti i prodotti software compatibili con MySQL. Il prodotto ha raggiunto dei livelli di maturità tali da rendere le istallazioni che lo usano molto stabili e affidabili.

La caratteristica architetturale di MySQL e MariaDB di separare il nucleo del server dallo Storage Engine, consente di scegliere tra funzionalità transazionali, che garantiscono una maggiore affidabilità, e una memorizzazione dei dati su disco non transazionale, che però è estremamente efficiente. Il meccanismo di immagazzinamento dei dati sfrutta infatti l'architettura "independent storage engine", consentendo la scelta del tipo di sistema di storage più appropriato alle necessità dell'architettura in cui il DBMS si colloca. Se, ad esempio, il sistema richiede un locking a livello di singolo record allora è possibile utilizzare il motore di storage XtraDB, completamente compatibile con lo storage di MySQL InnoDB. In caso di applicazioni che non richiedono il supporto di transazioni si può fare riferimento al sistema MyISAM per massimizzare le performance. MariaDB può essere usato anche in configurazione cluster per consentire la cosiddetta high-availability ovvero l'elevata affidabilità garantita da un'architettura ridondata.

MariaDB viene rilasciato con licenza open source GNU GPL v. 2.0.

Sesame

Sesame (http://www.openrdf.org/) consiste in un framework open source per processare in modo completo dati in formato RDF. In particolare fornisce le SAIL API (SAIL – Storage and Inference Layer) che permettono di astrarre l'interazione con lo strato di persistenza delle triple, affidato a specifici *storage engine*, e con i motori di gestione delle inferenze.

A livello applicativo consiste in un'applicazione Java Enterprise Edition da installare su un servlet container, come Apache Tomcat, e che memorizza i dati in strutture altamente ottimizzate, mantenute nel file system o in memoria. Sesame include anche un suo storage engine ma per realizzazioni più complesse che necessitano di performance significative, è consigliabile l'utilizzo in abbinamento con altri prodotti (es. la tecnologia Ontotext GraphDB Lite, la nuova versione del motore OWLIM, che è utilizzabile gratuitamente).

Sesame è completamente compatibile con la versione 1.1 del protocollo SPARQL: in particolare supporta anche l'estensione "SPARQL for update" che ha introdotto nel linguaggio dei costrutti per gestire l'aggiornamento dei dati del triple store.

Sesame viene distribuito con una licenza open, definita sulla base della BSD 3-Clause License.

Solr

Solr (http://lucene.apache.org/solr/ è un motore di ricerca testuale di classe enterprise, basato sulle librerie del progetto Lucene di Apache. È un prodotto open source duttile e scalabile, sviluppato in tecnologia Java e installabile in produzione su servlet container Java come Tomcat, Jetty o JBoss.

Fornisce delle API di accesso ai suoi servizi implementate attraverso dei web services HTTP che gestiscono dati in formato XML e JSON. La presenza delle interfacce web di amministrazione rende poi Solr semplice da gestire e da configurare.

Solr offre innumerevoli funzionalità applicative, tra cui:

- Scalabilità ed estendibilità dell'architettura software
- Ottimizzazione dei tempi di risposta tramite la funzione di query caching su protocollo HTTP. Garantisce inoltre eccellenti prestazioni e tempi di risposta contenuti a fronte di alti volumi di traffico.
- Raffinamento delle ricerche mediante faccette
- Evidenziazione dei termini (hit) coincidenti con la query di ricerca.
- Ricerca full-text tramite l'uso di operatori logici (AND, OR; NOT).
- Filtraggio dei campi su cui operare la ricerca e gestione delle tipologie speciali di campo (es. date e valori numerici). Solr supporta tipi di campo numerici e data e permette la gestione di essi mediante moduli speciali di conversione e filtraggio, ad esempio per la gestione dell'algebra delle date.
- Supporto multilingue (es. it, en, fr, es) aperto a ulteriori integrazioni linguistiche. Esistono persino delle estensioni per gestire testi in lingua latina.
- Moduli avanzati di analisi testuale, configurabili e personalizzabili.
- Configurazione di componenti aggiuntivi per l'analisi testuale avanzata che includono sistemi per la separazione delle parole (tokenizzazione), utilizzo di espressioni regolari, filtri basati su meccanismi di assonanza tra i termini (sounds like filters).
- Configurazione delle liste di stopword (parole da non indicizzare, dipendenti dalla lingua, ad esempio articoli, preposizioni, ecc.), di sinonimi e di parole protette
- Supporto delle funzioni di spelling suggestion, spelling correction, similarità tra documenti, scoring dei contenuti e gestione degli errori di battitura.
- Ottimizzazione del recupero dei dati sulla base della frequenza dei termini più ricercati.

Solr è distribuito con licenza open source Apache License v. 2.0.

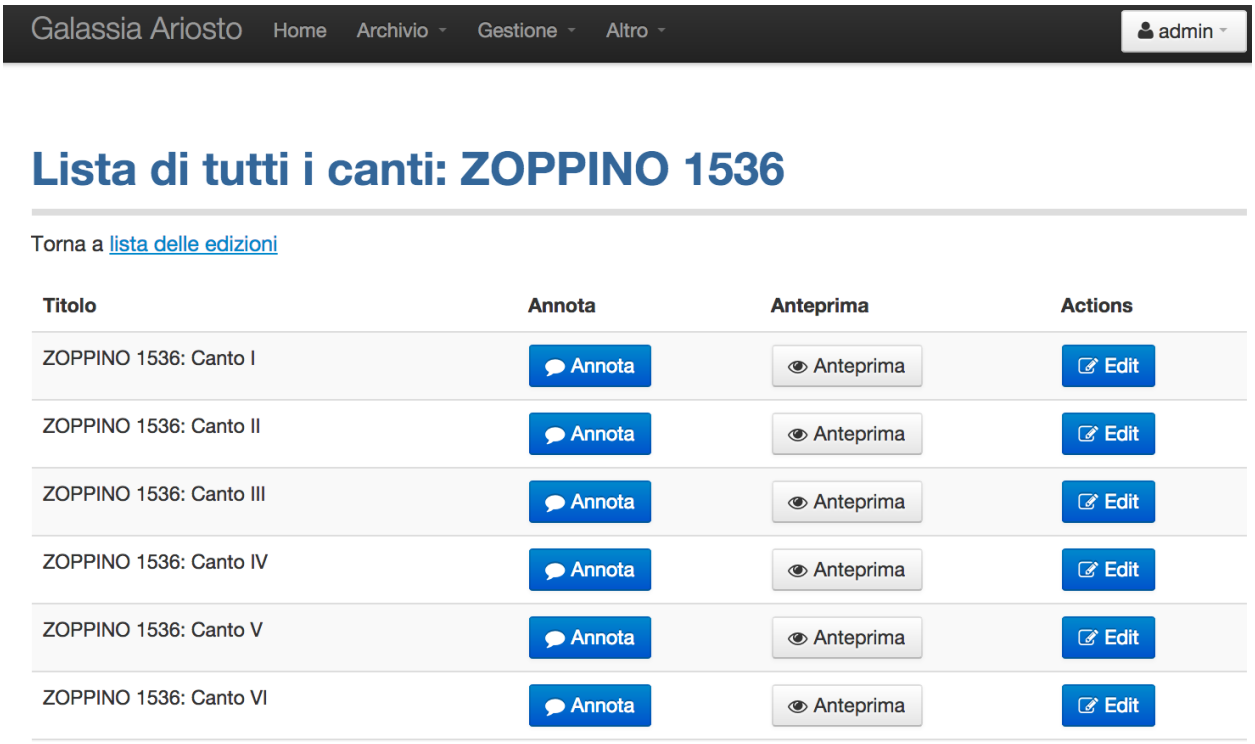
Wordpress (www.wordpress.com) è un software ampiamente utilizzato, con un mercato solido e rilasciato con licenza open source.

2.4 Le funzionalità principali del sistema

2.4.1 Gestione e fruizione di una biblioteca digitale di testi.

Attraverso la personalizzazione di uno dei componenti del framework MURUCA (http://www.muruca.org) verrà realizzato il modulo di interfaccia per l'utente editore.

Questo modulo si occupa del caricamento di file XML/TEI che vanno a costituire la biblioteca digitale. Al momento dell'upload il file viene interpretato da uno script che, attraverso un opportuno *mapping* tra metadati contenuti nel TEI header, e classi e predicati definite nei file OWL (si veda la sezione precedente), genera triple da salvare nel triplestore del "Data Layer". Lo stesso script si occupa della rappresentazione del testo caricato all'interno del DB relazionale che permette il salvataggio del testo in differenti formati: HTML per il frontend e HTML annotabile.
La conversione da XML/TEI nei differenti formati si basa sul software opensource oxGarage (http://www.oucs.ox.ac.uk/oxgarage/) che può essere utilizzato come servizio nella versione mantenuta dalla Oxford University o installato come pacchetto stand alone sul server che ospiterà tutti i moduli del progetto. oxGarage permette di utilizzare profili che, attraverso opportuni file XSLT, consentono la conversione di file XML/TEI in differenti formati (HTML, PDF, DOC, …).



La figura rappresenta un esempio di interfaccia per l'upload di file XML/TEI.

Un'operazione ulteriore che viene effettuata al momento dell'upload del file è l'indicizzazione dei metadati e del testo.

2.4.2 Front-end: Digital Library e Sito web

Nella soluzione proposta, si prevede la realizzazione di due siti distinti:

- la digital library, che sarà implementata tramite il framework Muruca,
- e il sito web del progetto, che sarà facilmente gestibile da utenti non esperti e consentirà la pubblicazione e l'aggiornamento di testi di presentazione del progetto.

Entrambi i sistemi si basano su tecnologie standard quali (X)HTML, CSS. Per entrambi, la presente offerta comprende un'interfaccia di base, progettata sulla base principi di semplicità e intuitività.

3.4.4. Motore e interfaccia di ricerca

Il motore di ricerca proposto, Solr, consente di fruire di un sistema di navigazione a faccette avanzato, che risponde ai requisiti di filtraggio indicati dal cliente (autore, genere, titolo, secolo, data, estensore), così come la ricerca testuale per parola, nome proprio o di luogo, gruppo di parole mediante operatori booleani, indici di prossimità e metacaratteri (caratteri jolly).

L'architettura proposta consente di implementare, in un secondo momento, caratteristiche extra quali:

- integrazione di un processore Saxon per la trasformazione XSLT e relativa resa statica HTML;
- integrazione di un motore di ricerca XML-enabled (XML-native).

2.5 Proprietà del software prodotto e modalità di rilascio

Il progetto farà uso di componenti open source, che manterranno la loro licenza. I codici sorgenti saranno rilasciati al committente, che potrà farne uso alle condizioni della licenza originale.

La proprietà del software prodotto ad hoc per il progetto è del committente, a cui saranno rilasciati i codici sorgenti senza alcun vincolo né relativo al suo uso, né relativo alla sua distribuzione, né di qualunque altro genere.

3. Piano di lavoro

Per lo sviluppo e la consegna del sito sono previsti **2 mesi** di lavoro, mentre il servizio di assistenza avrà una durata di 12 mesi a partire dal collaudo.

Si garantisce comunque che il sito sarà consegnato **entro il 30 giugno 2015**.
Il costo può essere suddiviso in fatturazioni separate.

4. Metodologie di lavoro

4.1 Strategie per il project management

Nello svolgimento delle attività di progetto sarà fondamentale l'approccio orientato alla qualità che Net7 intende proporre. L'azienda infatti ha il proprio Sistema di Qualità conforme e certificato ai requisiti della normativa UNI EN ISO 9001:2008: i suoi elementi distintivi verranno descritti nel presente paragrafo.

Viste le caratteristiche specifiche del presente appalto conviene evidenziare quegli aspetti metodologici del manuale di qualità che avranno maggiore rilevanza nel corso del progetto. In particolare i processi che saranno applicati per la presente fornitura sono:

- *Processo di Inizio*

  È la fase conseguente all'avvio ufficiale di un progetto, in cui si definisce l'organizzazione interna del team di lavoro e si stabiliscono le modalità operative per la conduzione delle attività. E' in questa fase che si devono gettare le buone fondamenta per garantire una qualità elevata nello svolgimento delle attività, minimizzare i rischi e massimizzare la soddisfazione del Cliente.

  In particolare in questa fase vengono svolte le attività di *pianificazione* del progetto, attraverso la definizione di due deliverable fondamentali: il Piano di Progetto e della Qualità e la Pianificazione Temporale del lavoro. Il primo definisce gli obiettivi principali di progetto, sia dal punto di vista tecnico, qualitativo che di marketing, e identifica l'organizzazione interna del team di lavoro necessaria al raggiungimento di questi obiettivi. La Pianificazione definisce il piano delle attività da eseguire nelle giuste sequenze temporali, fornendo per ciascuna di esse una stima della durata.

  Data la natura del presente appalto la Pianificazione Temporale si concentrerà soprattutto sulle fasi iniziali preparatorie del progetto. Le attività a richiesta invece non potranno in alcun modo essere pianificate in questa forma: per ciascuna di esse si definirà così un mini-piano temporale prima di avviare il lavoro.

- *Processo Tecnico*

  Definisce gli aspetti metodologici legati alla Qualità in base ai quali saranno svolte le attività di produzione nel progetto. Il processo tecnico quindi si focalizza sulla produzione dei deliverable necessari per *tracciare*, nello spirito della norma UNI EN ISO 9001:2008 gli aspetti fondamentali della produzione tecnica e si

sposa con gli standard di lavoro e le scelte tecnologiche seguite dal team e di cui daremo un'indicazione dettagliata nel paragrafo successivo.

In particolare il Processo Tecnico definisce che per ogni progetto devono essere previsti i seguenti documenti:

- *User & System Requirements*: identifica in maniera sistematica i requisiti del progetto, focalizzandosi sia sugli aspetti *Funzionali* che su quelli *Non-Funzionali* o di *Sistema*. I primi definiscono le caratteristiche che le realizzazioni del progetto devono avere e le funzionalità che devono soddisfare. I requisiti non-funzionali identificano le caratteristiche generali che le realizzazioni devono soddisfare e gli eventuali vincoli imposti dal contesto in cui si opera.
- *Architettura*: nel caso di progetti software questo deliverable permette di identificare nel dettaglio l'architettura hardware e software da implementare (o su cui operare). In particolare in questa fase verrà operata la scomposizione del sistema software in una serie di moduli, ognuno dei quali sarà poi oggetto di uno specifico sotto-processo realizzativo in cui saranno eseguiti i seguenti task:
  - *Analisi funzionale e progettazione di dettaglio*: definisce le specifiche tecniche e la progettazione del modulo.
  - *Sviluppo software*
  - *Test Unitari*, una pianificazione di test "unitari" relativi al moduli software in questione. Ogni attività di validazione basata su questi Unit Test dovrà essere documentata da appositi *Rapporti*, in cui viene riportato l'esito di ogni test effettuato.
- *Piano di Test Complessivo*: definisce i criteri di validazione generale del progetto. In particolare tre sono gli aspetti che verranno presi in considerazione per queste attività di verifica, ovvero:
  - *Test di Integrazione*: permettono di verificare se i moduli software sviluppati nel progetto (e testati nell'ambito del sotto-processo realizzativo) si integrano con successo
  - *Test Funzionali*: verificano se il sistema sviluppato soddisfa tutti i requisiti identificati nel documento di User & System Requirements
  - *Stress-Test*, il cui obiettivo è verificare che il sistema mantenga delle performance adeguate all'aumentare del "carico" di utenti contemporanei che lo utilizzano.

  L'esito delle verifiche effettuate saranno documentate in appositi *Rapporti di Test*.
- *Piano di collaudo*: è l'insieme delle prove, concordate con il Cliente e da eseguire nell'ambiente di produzione, che saranno effettuate durante i collaudi ufficiali del Progetto e stabiliti nel contratto. In seguito ad ogni

collaudo verrà rilasciato un *Rapporto* in cui saranno descritte nel dettaglio le prove eseguite e il relativo esito.

- *Documentazione di Installazione*: nei progetti di sviluppo software questo deliverable descrive nel dettaglio le operazioni di installazione e configurazione nonché tutte le attività di *fine-tuning* necessarie per garantire il funzionamento ottimale di un sistema in produzione. In particolare questo documento dovrà contenere indicazioni sulle modalità di installazione dell'hardware, del software di base, del software applicativo sviluppato e le operazioni da effettuare in eventuali sistemi esterni con cui il sistema dovrà interagire.

· *Processo di Controllo*

L'andamento del lavoro deve essere regolarmente monitorato e misurato per identificare eventuali scostamenti dal Piano di Progetto e dalla Pianificazione Temporale. Tale controllo è fondamentale e permette di prendere decisioni tempestive e appropriate per mantenere il progetto sempre in linea con i suoi obiettivi globali. In pratica, i processi di controllo hanno lo scopo di assicurare che il progetto:

- stia producendo i prodotti richiesti con il richiesto livello di qualità
- sia condotto nel rispetto dei piani di attività con i costi e le risorse concordate
- sia ancora valido rispetto al Business Plan approvato dal management.

I controlli o (*riesami*) dovranno essere esercitati con regolarità nell'intero ciclo di vita del progetto, in base alle politiche stabilite nel Piano di Progetto e della Qualità. Ogni riesame dovrà produrre una specifica registrazione, sia essa di tipo "formale" (es. un *Rapporto di Avanzamento Lavori*) o "informale" (ad esempio una mail inviata al team coinvolto nel progetto).

· *Processo di Garanzia*

Definisce le modalità con cui dovrà essere gestita la garanzia nel progetto nelle fasi di "post-rilascio", ovvero in seguito al collaudo con esito positivo del sistema. In particolare il processo si concentra su come dovrà essere svolta l'attività di manutenzione ordinaria, legata alla correzione degli errori di funzionamento del software applicativo o di comportamenti anomali dello stesso. Gli interventi relativi sono quindi volti a identificare e correggere tempestivamente i malfunzionamenti nella logica e nel funzionamento dei servizi che pregiudicano o rendono problematica l'utilizzazione delle funzionalità da parte degli utenti finali.

L'attività si basa su priorità di intervento definite in base alla criticità del componente funzionale del sistema coinvolto, al livello di gravità del problema e ai livelli di servizio che devono essere garantiti. Ogni intervento dovrà essere tracciato opportunamente attraverso la registrazione di un "ticket".

Una volta identificato il malfunzionamento ed analizzata la causa, viene verificata l'esistenza di una patch o sviluppate soluzioni correttive atte a ripristinare nel più breve tempo possibile l'operatività del sistema. Individuata ed implementata la soluzione correttiva, vengono eseguiti i test prima sull'ambiente di sviluppo del progetto e poi su quello di produzione, al fine di ottenere l'accettazione esplicita del Cliente e la sua autorizzazione a chiudere il ticket.

4.2 Strumenti e standard nell'analisi e nello sviluppo del software

La progettazione di sistemi informatici complessi necessita di metodologie professionali ed efficaci. A tale scopo il gruppo di lavoro di Net7 utilizzerà la metodologia *Unified Process* (UP). Si tratta di uno standard industriale basato su un modello "evolutivo", basato sulla prototipazione, fortemente influenzato dalle tecniche di sviluppo "agili" diffuse fra le comunità open source.

UP prevede uno sviluppo iniziale fino alla fase di primo rilascio e successivi continui sviluppi di versioni maggiormente "affinate" del prodotto finale (iterazioni). Le iterazioni coinvolgono generalmente le fasi di analisi dei requisiti, progettazione, sviluppo, test (unit e system test) e le verifiche di qualità. Un ciclo di vita "evolutivo" ha il vantaggio di garantire la disponibilità di un primo prodotto finito in tempi rapidi; di contro c'è il pericolo che il "prodotto finale" richieda un maggiore effort di sviluppo rispetto all'approccio "a cascata". Nel processo evolutivo assume pertanto una particolare rilevanza il controllo e la gestione dei rischi.

Il ciclo di vita del modello UP è diviso in diversi cicli, ognuno dei quali produce una nuova generazione di prodotto. Un ciclo di sviluppo è diviso in 4 fasi

- Fase iniziale (Inception);
- Elaborazione (Elaboration);
- Costruzione (Construction);
- Transizione (Transition).

Ogni fase a sua volta è suddivisa in iterazioni e ha una ben definita milestone che rappresenta un obiettivo da raggiungere.

La definizione formale di un processo di sviluppo UP è stata basata su alcune best practice, elencate di seguito, che nel tempo hanno rivelato tutta la loro importanza per la realizzazione di progetti software complessi:

- *Sviluppo iterativo del software*. UP utilizza un approccio iterativo allo sviluppo del software: al termine di ciascuna iterazione, è previsto il rilascio di una versione parziale, ma funzionante, del software che si intende realizzare. La scelta dell'approccio iterativo, rispetto ad un approccio sequenziale, viene giustificata dall'impossibilità, nei moderni progetti di sviluppo, di definire preventivamente tutti i requisiti necessari a costruire compiutamente il sistema desiderato. Questo approccio consente inoltre di affrontare, nelle prime iterazioni, le parti

considerate ad alto rischio per il progetto, consentendo in questo modo di eliminare per tempo i problemi funzionali ed architetturali.

- *Orientamento ai requisiti*. UP è un processo guidato dai requisiti. Attraverso la stesura di use case e di scenari, vengono identificati i requisiti funzionali i quali guidano poi tutte le fasi di realizzazione, dall'analisi alla progettazione, dall'implementazione al test.
- *Sviluppo del software basato su componenti*. UP è un processo di sviluppo orientato alla realizzazione di sistemi object-oriented distribuiti, utilizzando un'architettura basata su componenti software per garantire la massima riutilizzabilità e modularità del codice prodotto.
- *Modellazione visuale del software*. UP prevede, in tutte le fasi di sviluppo, l'utilizzo di UML (Unified Modeling Language), per progettare modelli visuali del sistema, a differenti livelli di profondità. L'utilizzo di UML garantisce da un lato la consistenza tra le fasi di progettazione e di implementazione, e dall'altro facilita la comunicazione tra le diverse figure professionali impegnate nel progetto.
- *Qualità del software*. UP pone una particolare enfasi sulla progettazione, in tutte le fasi del progetto, di una serie di test di qualità, allo scopo di poter pianificare e verificare da subito l'aderenza del codice prodotto rispetto ai requisiti.
- *Change management*. UP descrive come controllare e tracciare i cambiamenti, allo scopo di poter eseguire con successo le iterazioni previste nello sviluppo di un progetto.

Punti di forza nell'utilizzo di questa metodologia sono:

- flessibilità del ciclo di vita;
- centralità degli aspetti architetturali;
- uso di tecnologie Object-oriented;
- focus sui rischi del progetto e sulla loro gestione.

Le quattro fasi sono descritte in dettaglio nel seguito:

- *Inception*: Durante questa fase, viene eseguito lo studio di fattibilità del progetto, in modo da evidenziare con chiarezza lo scopo del progetto stesso. Durante la fase di inception viene prodotto un documento di "Vision" che illustra, in termini generali, le caratteristiche chiave ed i requisiti del componente software che si intende realizzare. Viene anche prodotto uno use case model iniziale, il quale ha lo scopo di evidenziare i principali requisiti funzionali del sistema. La milestone di questa fase prevede come criteri di valutazione l'accettazione, da parte del cliente e del team di sviluppo, di quanto definito durante la fase, in termini di scopo, qualità, budget e tempo previsti.
- *Elaboration*: Durante questa fase viene analizzato il dominio del problema e, contemporaneamente, viene definita e realizzata l'architettura sulla quale il sistema verrà implementato. È importante notare che, durante questa fase, oltre alla raccolta sistematica e dettagliata della maggior parte degli use case, viene progettato lo strato di software architetturale, ritenuto idoneo ai fini del sistema. Fondamentale è inoltre in questa fase l'analisi dettagliata del modello dei dati e l'elaborazione di una proposta per definire la loro struttura nel progetto. La milestone di questa fase prevede come criteri di valutazione la consistenza degli

use case prodotti (circa l'80% del totale), e la corretta progettazione dello strato architetturale del sistema e del modello dei dati.

- *Construction*: E' la fase dello sviluppo del progetto, in cui le feature del sistema vengono implementate in una serie di brevi iterazioni timeboxed. Al temine di ogni terazione viene rilasciata una nuova versione del software. I casi d'uso, descritti in forma di storie, sono il punto d'inizio di ogni iterazione.
- Transition: E' l'ultima fase del processo. Il software viene rilasciato per essere testato dagli utenti finali. I feedback ricevuti vengono recepiti e possono dar luogo a modifiche del software, oggetto di ulteriori iterazioni. In questa fase può essere prevista la formazione agli utenti.

Le attività che si svolgono durante le quattro fasi del processo, possono essere raggruppate in due gruppi: le attività di processo (Process Workflows), più strettamente legate al ciclo di vita del software e le attività di supporto (Support Workflows), più orientate al management del progetto. Diamo di seguito una descrizione di queste due tipologie di workflow.

Fra i workflow di processo quelli che riguarderanno le attività di progettazione del sistema software saranno:

- *Business Modeling*: Questo workflow consiste nella modellazione concettuale di un processo di business, attraverso la comprensione dell'organizzazione e la definizione dei requisiti del sistema. Il risultato di questa attività costituisce una progettazione esecutiva, che comprende anche il modello concettuale dei dati e i cosiddetti business use case, o use case di alto livello.
- *Requirements*: Questo workflow consiste nel descrivere i requisiti funzionali e non funzionali del sistema mediante il loro raffinamento e formalizzazione in un linguaggio che sia contemporaneamente comprensibile da parte degli utenti finali e degli sviluppatori. Gli artefatti principali prodotti in questo workflow sono gli use case, e lo use case model UML. Viene prodotto anche il modello logico dei dati e il piano di massima dei system test. Alla conclusione di questa fase, è prevista una sessione di verifica e validazione interna dei risultati e del livello di soddisfacimento dei requisiti.
- *Analysis & Design*: Scopo di questo workflow è quello di mostrare un modello di soluzione dei requisiti rilevati allo step precedente del processo. Esso prevede la formalizzazione tecnica dei requisiti e dell'architettura tecnologica. In merito alla formalizzazione tecnica dei requisiti, in questa fase vengono prodotti i sequence diagram per i casi d'uso più complessi.

5. Offerta economica

Il costo previsto per la realizzazione della Biblioteca digitale ALIM2 è 10.000 Euro (IVA esclusa).

La presente offerta include 12 mesi di hosting dell’applicazione su server cloud che possiede tutti i requisiti tecnici necessari al funzionamento dell’applicazione e 12 mesi di manutenzione e assistenza.

6. Garanzia

Si garantisce il software sviluppato per un anno dalla data del collaudo definitivo. La garanzia comprende:

- assistenza telefonica e per mail, in orario lavorativo (help-desk di secondo livello);
- esame e soluzione di problemi segnalati nel periodo;
- aggiornamenti tecnologici della piattaforma.