ANAC
Autorità Nazionale Anticorruzione

Allegato 2

Linee guida tecniche per lo sviluppo dei Servizi dell'ANAC

2.0

Indice

1.1 Dati del documento

Nome documento	**Linee guida tecniche Servizi ANAC**		
Versione documento	2.0	**Codice documento**	Servizi-A.N.AC.-LGT
Data creazione documento	09.02.2012	**Data ultimo aggiornamento**	07.10.2015

1.2 Definizioni e acronimi

Termine/Acronimo	Descrizione
AVCP	Autorità per la Vigilanza Contratti Pubblici Organo collegiale che vigila sul rispetto delle regole che disciplinano la materia dei contratti pubblici, dotata di indipendenza funzionale, di giudizio, di valutazione e di autonomia organizzativa.
ANAC	Autorità Nazionale Anticorruzione
Autorità	Si riferisce all'ANAC
Struttura Committente (SC)	
UAFI	Ufficio Analisi Flussi Informativi
UPSI	Ufficio Progettazione e sviluppo, Servizi Informatici e Gestione del Portale dell'ANAC.
UESI	Ufficio Esercizio Sistemi
BDNCP	Banca Dati Nazionale dei Contratti Pubblici Anagrafe unica dei contratti pubblici. È la banca dati di riferimento di ANAC per utenze e soggetti rappresentati.
Portale Internet	Portale Internet Punto di erogazione dei servizi web per gli utenti esterni ANAC
Servizi ANAC	Servizi ANAC Nucleo centralizzato di servizi che contengono la logica di Business

	dell'Autorità.
SPC	Sistema Pubblico di Connettività L'insieme di infrastrutture tecnologiche e di regole tecniche per lo sviluppo, la condivisione, l'integrazione e la diffusione del patrimonio informativo e dei dati della pubblica amministrazione.
PDD	Porta di dominio
FWXML	Firewall XML
B2C	Business to Consumer
B2B	Business to Business
SOA	Service Oriented Architecture
LDAP	Lightweight Directory Access Protocol
ESB	Enterprise Service BUS
KMS	knowledge management system
EIS	Enterprise Information System
RPC	Responsabile della prevenzione della corruzione
RT	Responsabile della Trasparenza
RUP	Responsabile Unico del Procedimento
RASA	Responsabile anagrafica stazione appaltante
OE	Operatore economico

Tabella 1 - Definizioni e acronimi

1.3 Convenzioni di carattere generale

Identificativo	Descrizione
RIF. {short_desc}.nnn	Identificativo univoco di un riferimento dove {short_desc} è una descrizione breve e nnn è un progressivo numerico
REQ.{short_desc}.nnn	Identificativo univoco di un requisito dove {short_desc} è una descrizione breve e nnn è un progressivo numerico

Tabella 2 - Convenzioni

1.4 Contesto normativo

Codice riferimento	Riferimento normativo	Descrizione
RIF.NORM.01	Legge 9 Gennaio 2004, n.4	Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici" (c.d. "Legge Stanca")
RIF.NORM.02	d.P.R. 1 marzo 2005, n.75	Regolamento di attuazione della Legge 9 gennaio 2004, n. 4, per favorire l'accesso dei soggetti disabili agli strumenti informatici
RIF.NORM.03	D.M. 8 luglio 2005	Requisiti tecnici e i diversi livelli per l'accessibilità agli strumenti informatici
RIF.NORM.04	D.M. 20 marzo 2013	Aggiornato l'allegato A del D.M. 8 luglio 2005 in merito ai requisiti tecnici e i diversi livelli per l'accessibilità agli strumenti informatici. Si tratta di un aggiornamento dei requisiti di accessibilità informatica dalle WCAG 1.0 alle WCAG 2.0
RIF.NORM.05	Circolare n.61/2013 del 29 marzo 2013 dell'Agenzia per l'Italia Digitale	Accessibilità dei siti web delle pubbliche amministrazioni
RIF.NORM.06	D.Lgs. 7 marzo 2005, n.82 smi	Codice dell'Amministrazione digitale
RIF.NORM.07	Direttiva 27 luglio 2005	Qualità dei servizi on line e misurazione della soddisfazione degli utenti emanata dal Ministro per l'innovazione e le tecnologie e dal Ministro per la funzione pubblica

Tabella 3 - Contesto normativo

1.5 Riferimenti interni

Codice riferimento	Descrizione del documento	Versione	Codice del documento

RIF.DOCINT.001	Linee guida interfaccia grafica Portale AVCP	1.0	DOC_Linee_interfaccia_grafica_ANAC
RIF.DOCINT.002	IAM - Linee guida per integrazione applicazioni	1.0	IAM - Linee guida per integrazione applicazioni – v1.0

Tabella 4 - Riferimenti interni

1.6 Riferimenti esterni

Codice riferimento	Codice	Descrizione
RIF.EXT.001	ISO/IEC 17799	Codice di condotta per la gestione della sicurezza dell'informazione
RIF.EXT.002	ISO/IEC 25010 (ISO/IEC 9126).	Standard di qualità software
RIF.EXT.003	ISO/IEC 12207	Gestione del ciclo di vita del software
RIF.EXT.004	"Enterprise Integration Pattern"	2011 - Gregor Hohpe e Bobby Woolf. Designing, Building, and Deploying Messaging Solutions
RIF.EXT.005	"SOA design pattern"	2009 – Tomas Erl
RIF.EXT.006	Java Code style	Google - https://google.github.io/styleguide/javaguide.html
RIF.LGS.01	Linee guida per i siti web della PA	Emanate da DigitPA (ora Agenzia per l'Italia Digitale) nel 2011, in riferimento all'art. 4 della Direttiva n. 8/2009 del Ministro per la pubblica amministrazione e l'innovazione
RIF.LGS.02	SPCoop – Servizio pubblico di cooperazione applicativa	Riferimento tecnico per la cooperazione applicativa fra le amministrazioni pubbliche

RIF.LGS.03	Linee guida nazionali per la valorizzazione del patrimonio informativo pubblico	Documento di LINEE GUIDA per l'individuazione degli standard tecnici, compresa la determinazione delle ontologie dei servizi e dei dati, le procedure e le modalità di attuazione delle disposizioni del Capo V del Codice dell'Amministrazione Digitale con l'obiettivo di rendere il processo omogeneo a livello nazionale, efficiente ed efficace

Tabella 5 - Riferimenti esterni

1.7 Versioni del documento, validità e processo di aggiornamento

L'aggiornamento delle linee guida è di specifica competenza dell'Autorità, la quale si riserva di cambiare in qualsiasi momento il contenuto del presente documento. Ogni variazione viene tracciata nel presente paragrafo. Nella tabella seguente vengono riportate le versioni del documento; la data di emissione, coincidente con la pubblicazione del documento e con la comunicazione ai fornitori interessati; la data di entrata in vigore delle nuove linee guida, che di norma avviane 30 giorni dopo la loro emissione, in casi particolari è possibile specificare una data di entrata in vigore differente.

Dalla data di entrata in vigore il fornitore è tenuto a rispettare la nuova versione delle linee guida

Versione	Data emissione	Data di entrata in vigore	Descrizione delle modifiche apportate
1.0	18/07/2012	18/07/2012	Prima stesura
1.1	31/07/2012	31/07/2012	Uniformata versione EPP 5.2.1 Indicato riferimento linee guida IAM
1.2	08/08/2012	08/08/2012	Armonizzazione dei layer applicativi
1.3	28/08/2012	28/08/2012	Formato messaggio LOG Requisito non funzionale file di configurazione Piattaforma di testing refusi

2.0	07/10/2015	07/10/2015	Revisione totale del documento

Tabella 6 - Versioni del documento

1.8 Indice delle figure

2 Introduzione

2.1 Obiettivo del documento

Lo scopo del presente documento è di definire il modello di riferimento per l'intero ciclo di vita dei prodotti software realizzati in Autorità al fine di semplificare le fasi di analisi e progettazione standardizzando sia le tecniche progettuali e metodologiche, sia le fasi operative di esercizio.

A tal fine verrà presentata l'architettura di riferimento orientata ai servizi dell'Autorità, i processi da seguire nel ciclo di vita del software e le linee guida per lo sviluppo delle applicazioni.

2.2 Struttura del documento

Il presente documento è strutturato in 3 capitoli:

- Architettura di riferimento dell'Autorità, Capitolo 3
- Ciclo di vita del software, Capitolo 4
- Linee guida per lo sviluppo delle applicazioni, Capitolo 5

2.3 Concetti di base

Di seguito vengono illustrati i concetti, le tecnologie e le metodologie di riferimento adottati nella realizzazione dell'architettura logica e fisica dei sistemi dell'Autorità.

2.3.1 Service Oriented Architecture

Con SOA si identifica un paradigma architetturale per la progettazione, realizzazione e governance di architetture distribuite, in cui la collaborazione tra sistemi avviene tramite lo scambio di servizi. In tale ottica, ogni servizio, indipendentemente da come questo venga realizzato, eroga funzionalità tramite delle interfacce ben definite e a sua volta può utilizzare funzionalità messe a disposizione da altri servizi, realizzando quindi applicazioni di maggiore complessità. La SOA, che identifica una particolare forma di sistema distribuito, non è quindi legata ad una specifica tecnologia o insieme di tecnologie, bensì è una strategia IT che modifica sostanzialmente il modo di pensare e mettere in opera le attività e le organizzazioni di business. Definisce inoltre una serie di proprietà, orientate al riutilizzo e all'integrazione in un ambiente eterogeneo, che devono essere rispettate dai servizi che compongono il sistema. L'output atteso, dall'introduzione della SOA, è una collezione di servizi condivisi, standardizzati e governati in maniera indipendente, il cui scope va oltre la realizzazione del singolo processo di business in quanto si estende a più processi di business.

2.3.2 Application lifecycle management

Application Lifecycle Management (ALM) identifica un approccio strategico alla gestione delle informazioni, dei processi e delle risorse a supporto del ciclo di vita del software, dalla loro ideazione, allo sviluppo, alla messa in esercizio (produzione). L'ALM non è una tecnologia informatica, ma piuttosto un approccio integrato, basato su un insieme di tecnologie, su metodologie di organizzazione del lavoro collaborativo e sulla definizione di processi.

L'obiettivo dell'ALM è ottimizzare (minor tempo, minori costi, maggiore qualità, minori rischi) il design, lo sviluppo e l'esercizio delle applicazioni e sistemi software. L'obiettivo di "software delivery" si realizza come ciclo di attività strettamente relazionate tra loro: definizione, design, sviluppo, test, deployment e management. Ognuno di questi elementi deve essere attentamente gestito e controllato.

La proposizione delle tecniche di Application Lifecycle Management guida la soluzione tecnologica all'implementazione permettendo il conseguimento dei seguenti effetti e benefici agli sviluppi:

- incremento della produttività, attraverso la condivisione di best practices per sviluppo e deployment, e focalizzazione esclusiva sui business requirements da parte del gruppo di sviluppo;
- incremento della qualità finale dei deliverables, in modo tale da soddisfare necessità e aspettative degli utenti finali;
- superamento dei vincoli di comunicazione tra gruppi grazie a maggiore collaboratività e flusso d'informazioni più agile;

- accelerazione del processo di sviluppo con tecniche di integrazione semplificata;
- riduzione dei tempi di manutenzione grazie alla sincronizzazione sviluppi e analisi;
- massimizzazione degli investimenti negli skills, processi, tecnologie;
- incremento della flessibilità riducendo il tempo richiesto per le operazioni di build e adattando le applicazioni che supportano nuove specifiche di business.

2.3.3 Layering

Il concetto di stratificazione (layering) comporta la scomposizione del sistema in sottosistemi di minore complessità l'identificazione di una serie di strati logici separati. Ogni strato, nell'ottica di realizzare le proprie funzionalità, usufruisce degli strati di più basso livello e mette a disposizione servizi agli strati superiori.

Nella figura seguente è mostrata la suddivisione in strati logici che costituisce il modello di riferimento per lo stack applicativo dell'Autorità.

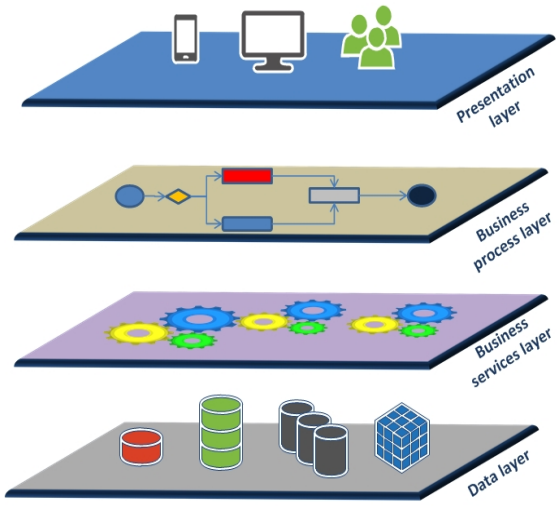

Figura 1 - Architettura multi-layers

2.3.4 Clustering & Balancing

Al termine Cluster può essere attribuito un significato formale: un cluster è un gruppo di risorse finalizzate ad un obiettivo comune, e sono consapevoli gli uni degli altri. Il clustering prevede la configurazione delle risorse (server di solito) per la comunicazione di informazioni su un particolare canale (porta) attraverso la quale possa avvenire lo scambio dei loro stati, quindi lo stato di una risorsa viene replicato in altri luoghi. Il clustering di solito include anche il bilanciamento del carico, in cui la richiesta è indirizzata ad una delle risorse del cluster secondo la politica di bilanciamento prevista. Il bilanciamento del carico può anche avvenire senza il clustering. In questo caso abbiamo più server

indipendenti ma con la stessa configurazione, i quali non sono a conoscenza l'uno degli altri. Un bilanciatore di carico inoltra le richieste ad uno dei server. Ogni server non utilizza le risorse degli altri server e non condivide il suo stato con altre risorse. Il bilanciatore del carico fondamentalmente esegue una sola attività: e fare in modo e controllare continuamente che i server siano attivi. Quando viene ricevuta una nuova richiesta, viene inviata ad uno dei server secondo la politica di bilanciamento del carico. Quando si riceve una richiesta per un utente che ha già una sessione, questa viene inoltrata allo stesso server che l'ha generata (questa parte è importante, altrimenti un utente potrebbe essere re-diretto verso i diversi server, che non sarebbero in grado di elaborare la richiesta).

3 Architettura di riferimento dell'Autorità

In questo capitolo viene presentata una visione d'insieme del disegno logico e fisico dell'architettura 'service-centric' adottata da parte dell'Autorità per garantire l'erogazione di:

- servizi applicativi da fornire agli stakeholder esterni all'Autorità come stazioni appaltati, operatori economici, enti pubblici e privati;
- servizi di cooperazione applicativa su SPCoop;
- insieme di applicazioni aziendali interne all'Autorità;
- insieme di servizi infrastrutturali;
- insieme di servizi trasversali.

3.1 Caratteristiche generali

L'architettura generale di riferimento dell'Autorità è stata realizzata secondo il pattern architetturale orientato ai servizi (SOA) presentato in precedenza, con lo scopo di sfruttarne i vantaggi offerti. Al fine di garantire l'aderenza ai requisiti fondamentali di un'architettura multi-layer si è proceduto a:

1. realizzare un layer di presentation standardizzando l'interfaccia utente per le applicazioni web-based;
2. separare e incapsulare la logica di business in specifici layer con compiti di carattere generale o specifico;
3. isolare il livello di accesso ai dati.

3.2 Tipologia dei servizi erogati

I servizi erogati dall' Autorità sono sostanzialmente di quattro tipologie:

- **Servizi B2C**: sono i servizi che consentono di interagire con l'utente finale, sia interno sia esterno all'Autorità, attraverso un interfaccia web fruibile tramite un browser.
- **Servizi B2B**: sono i servizi che consentono l'interoperabilità e la cooperazione (puntuale o massiva) tra i sistemi previo accordo di servizio (di norma attraverso il framework SPCOOP) o protocollo d'intesa.
- **Servizi Trasversali**: sono i servizi di supporto all'erogazione dei servizi dell'Autorità come servizi di gestione documentale, sicurezza applicativa, service registry e repository dei servizi.
- **Servizi Infrastrutturali**: sono i servizi che compongono l'infrastruttura dell'Autorità come servizi di rete, sistema, data center, sicurezza, posta elettronica, firma digitale e monitoraggio della qualità dei servizi.

Nella figura seguente è mostrato il modello concettuale e gli stakeholder dei servizi erogati dall'Autorità:

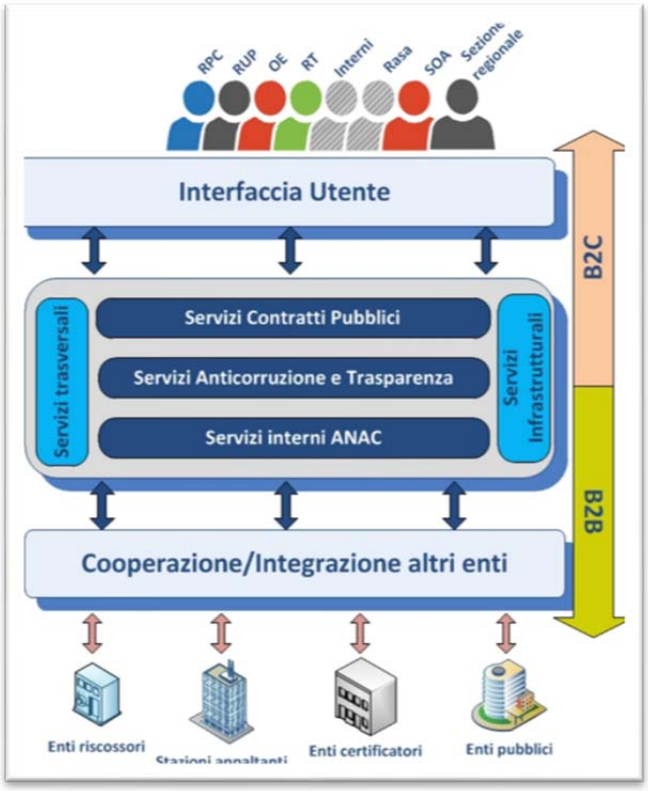

Figura 2 - Modello concettuale dei servizi erogati dall'Autorità

3.3 Tipologie di utenti

I servizi sono accessibili da varie tipologie di utenti:

1. Utenti interni dell'Autorità che possono:

a. Consultare i dati attraverso un portale web oppure attraverso funzioni ad-hoc;
b. Utilizzare i servizi gestionali dell'Autorità;
c. Accedere alla intranet aziendale.

2. Utenti esterni dell'Autorità registrati e profilati per l'utilizzo dei servizi ad accesso riservato come SIMOG, RISCOSSIONE, SMARTCIG, etc;
3. Utenti esterni dell'Autorità, per l'accesso pubblico ai servizi come Catalogo dei dati aperti, Portale trasparenza, casellario delle imprese qualificate, etc;
4. Enti esterni, per l'accesso ai servizi in cooperazione applicativa o mediante integrazione di sistemi.

3.4 Architettura logica

Questo paragrafo descrive l'attuale architettura applicativa del sistema informativo dell'Autorità e del percorso di evoluzione da completare per arrivare ad una architettura del tutto omogenea e orientata ai servizi.

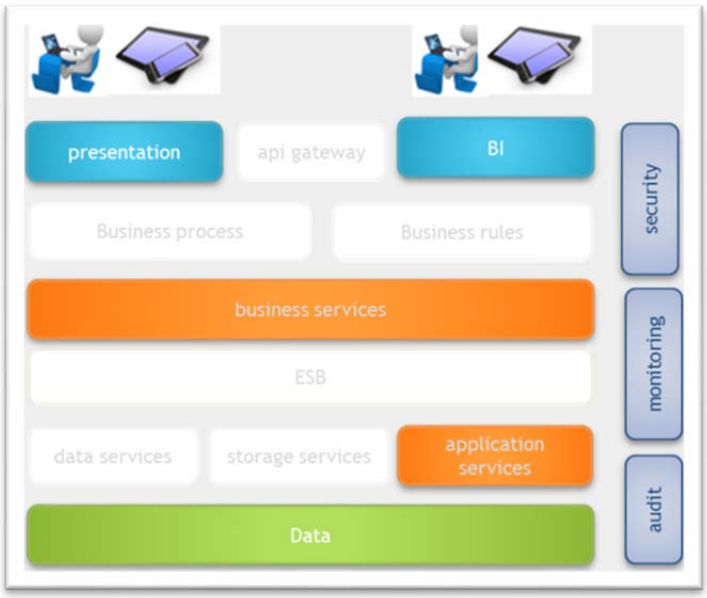

Figura 3 - Architettura Logica AS-IS vs TO-BE

Lo schema riporta le componenti previste nell'architettura TO-BE, con evidenza nei riquadri colorati delle componenti attualmente presenti.

3.4.1 Presentation Layer

Questo capitolo descrive le linee guida fondamentali per la progettazione del livello di presentazione di un'applicazione e le tematiche principali da affrontare durante la fase di progettazione.

Questo strato applicativo contiene i componenti che implementano l'interfaccia utente e ne gestisce l'interazione in ottica B2C multicanale.

Ogni funzionalità da implementare viene resa disponibile in un contesto intranet o internet, in base al tipo di utenza che ne fruisce, come indicato nella seguente figura.

Figura 4 Contesto internet-intranet dei servizi dell'ANAC

Questa separazione consente un miglior livello di sicurezza attraverso la definizione di policy di accesso differenziate. Inoltre, consente di avere punti unici di accesso ai servizi con interfacce utente personalizzate per la classe di utente raggiunta.

3.4.1.1 Servizi esterni (Internet)

Sono i punti di accesso per gli utenti esterni (Stazioni appaltanti, Operatori economici e Società organismo di attestazione, etc) verso i servizi erogati dall'Autorità. Il *presentation layer* di ogni servizio erogato, esposto agli utenti esterni attraverso il canale web, è integrato nel portale oppure è realizzato come componente esterno ma comunque sempre accessibile dalla sezione 'Servizi' del portale istituzionale. Questa sezione si suddivide in una area ad accesso libero, fruibile da tutti gli utenti, ed in un area ad accesso riservato che richiede un accesso profilato. Le specifiche tecniche di integrazione delle applicazioni con le componenti relative all'autenticazione e alla profilazione sono riportate nell'apposito documento (**RIF.DOCINT.002**). Il portale, compatibile con le specifiche JSR168 e JSR286, consente all'utente autorizzato di accedere alla sua *dashboard* personale, punto di partenza verso le funzionalità del portale per l'elaborazione di contenuti e layout personalizzati. Ogni funzionalità condivide lo stesso layout grafico e le regole definite nel documento "Linee guida interfaccia grafica Portale AVCP" (**RIF.DOCINT.001**).

Fanno parte dei servizi internet, messi a disposizione degli utenti:

- **Portale Istituzionale**: contiene le informazioni, attività, orientamenti e comunicazioni dell'Autorità, inoltre i riferimenti e le informazioni sui servizi e la parte di "Amministrazione Trasparente". Tecnologie utilizzate eXo Portal (RIF.PROD.008);
- **Portale dei servizi**: contiene i servizi informatici integrati nel portale. Tecnologie utilizzate Jboss EPP (RIF.PROD.004);
- **Servizi Web**: contiene i servizi informatici non integrati nel portale. Tecnologie utilizzate Jboss EAP (RIF.PROD.003);
- **Catalogo dati aperti:** portale ad accesso libero contenente i dati in formato digitale standard aperto. Tecnologie utilizzate AngularJS (RIF.PROD.009) e JSON.

A tendere, tale strato dovrà essere evoluto anche nell'ottica di avere un unico portale dei servizi dell'Autorità.

3.4.1.2 Servizi interni (Intranet/Extranet)

Sono i punti di accesso verso i servizi per gli utenti interni dell'Autorità. Attualmente è presente una intranet aziendale contenente le informazioni sulle News, la struttura organizzata dell'Autorità, i servizi Informatici, gli strumenti di lavoro e le aree di collaborazione realizzata con tecnologia Sharepoint (RIF.PROD.010). Inoltre, sono presenti altri servizi web-based realizzati con diverse tecnologie (php, Java, .net). A tendere, tale strato deve essere evoluto, ove possibile, nell'ottica di avere un unico portale dei servizi interni dell'Autorità.

3.4.2 Cooperation Layer

Il layer, al fine di rendere disponibile i servizi erogati dall'Autorità ad Enti esterni e viceversa, espone interfacce per la comunicazione con sistemi esterni (business to business – b2b) al fine di instaurare domini di cooperazione, anche secondo le specifiche di cooperazione SPCoop qualora siamo implementate dall'ente erogatore/fruitore.

Le principali componenti presenti al suo interno sono:

1. una PDD conforme alle specifiche SPCoop per la comunicazione verso l'esterno;
2. un firewall XML;
3. servizi di business esposti sia su canale SPCoop che su altri canali diretti.

L'evoluzione di tale strato è di delegare ad un servizio dell'ESB, invocato sia dall'interno che dall'esterno, la conoscenza dei dettagli ed informazioni per eseguire la comunicazione in modalità definite dalla specifica.

3.4.2.1 *Cooperazione applicativa su canale SPCoop*

La cooperazione applicativa su canale SPCoop viene effettuata mediante l'esposizione di servizi SOAP attraverso la porta di dominio, interfaccia software tra due sistemi per il consumo di servizi esposti. Ad oggi, i servizi interni che necessitano di invocare i servizi esterni in cooperazione applicativa su canale SPCoop devono interfacciarsi direttamente con la PDD e viceversa.

I servizi devono essere esposti secondo le specifiche definite nel documento RIF.LGS.02.

3.4.2.2 *Cooperazione applicativa diretta*

Nel caso altri Enti esterni non siano predisposti all'erogazione dei servizi in modalità SPCoop, l'Autorità, al fine di consentire comunque l'esposizione del servizio, consente la creazione di domini in trust, anche mediante reti VPN e mediati da firewall xml per garantirne l'accesso secondo le policy di sicurezza definite per il servizio.

I servizi devono essere esposti secondo le specifiche definite nel documento RIF.LGS.02.

3.4.2.3 *Sistemi di interscambio dei flussi di dati*

Nel caso sia necessario effettuare scambio di dati in modalità massiva, l'Autorità consente lo scambio di file, contenenti tali dati, su canale FTP sicuro. I formati maggiormente utilizzati per tale scambio dati sono l'XML, JSON e CSV, MDB. Possono essere aggiunti altri formati in base alle specifiche esigenze dell'Autorità.

3.4.3 Business process

E' lo strato che implementa i workflow utilizzati nei processi di business, garantendo anche la gestione dello stato dei processi in modo persistente.

L'Autorità a tal fine ha individuato ed acquisito la suite JBoss BPM (RIF.PROD.006) per la realizzazione dei processi e le operazioni funzionali al completamento degli adempimenti previsti sia da parte degli utenti esterni che interni all'Autorità. Attraverso tale componente è possibile modellare, secondo lo standard BPMN 2.0, i processi dell'Autorità basati sui servizi di business mediati dalla componente ESB presente nel layer di integrazione.

3.4.4 Business rules

E' lo strato che centralizza le regole di business, definite attraverso un metalinguaggio, in un unico repository e ne garantisce una gestione semplificata anche attraverso il versionamento e la validità temporale delle stesse. L'Autorità a tal fine ha individuato ed acquisito la suite JBoss BRMS

(RIF.PROD.007). Le regole dovranno essere realizzate in base alle esigenze nelle modalità previste dalla piattaforma.

3.4.5 Business service

Si occupa di eseguire i servizi di business dell'Autorità, cuore dell'architettura applicativa, necessari per l'esecuzione delle funzioni elementari. I servizi principali di tale strato sono i servizi BDNCP, componenti che realizzano le principali funzionalità per gestire le aree core dell'Autorità come appalti, casellario, anagrafe. La logica di accentrare in un unico punto le funzionalità di business chiamate da più consumer, consente di migliorarne la gestione e contenere gli impatti, evitando quindi duplicazioni e consentendo una flessibilità maggiore nei cambiamenti. La tecnologia di riferimento per la realizzazione dei servizi e EJB 3.0. Tali servizi verranno mediati dallo strato di integration ed esposti in modalità sincrona e asincrona nei formati previsti (EJB, WS, REST, JMS, FTP, ecc).

3.4.5.1 Integration Layer

Lo strato di integrazione costituisce un elemento fondamentale dell'architettura dell'Autorità. Consente di utilizzare, attraverso diversi tipi di interfacce di comunicazione, i servizi invocati attraverso l'interfaccia web o attraverso lo strato di cooperazione. La tecnologia utilizzata per realizzare tale strato è un ESB, interfaccia unica attraverso cui transita ogni richiesta di comunicazione tra i servizi e le applicazioni che devono interoperare tra loro. Si occupa di esporre i servizi di business con diverse interfacce standard o adapter in maniera da integrare sistemi eterogenei offrendo anche tutti i servizi infrastrutturali di un middleware quali routing, data mapping, sicurezza, trasformazione dei dati, etc….
L'Autorità a tal fine ha individuato ed acquistato la suite JBoss Fuse (RIF.PROD.005), basata su architettura OSGI, che consente di implementare i servizi attraverso i pattern di integrazione Camel. L'integrazione dei servizi da erogare deve essere progettata e realizzata secondo i pattern di integrazione definiti nel testo "Enterprise Integration Pattern" (RIF.EXT.009).

3.4.6 Data Layer

Rappresenta l'ultimo strato dell'architettura, occupandosi di gestire i dati resi persistenti su database e file system.

3.4.6.1 Data service

Sono i servizi, messi a disposizione della logica applicativa, che si occupano di conservare e gestire le informazioni attraverso il Database Management System. L'accesso ai dati è realizzato, oltre che con le tecnologie ORM (Hibernate, MyBatis) anche attraverso *store procedure*, *function* e *viste* messe a disposizione

dei servizi per l'accesso e gestione dei dati. Sono presenti anche componenti ETL per la gestione dei dati tra le varie banche dati in modo da consentirne l'integrazione, trasformazione e controllo della qualità.

La suite DBMS adottata è Microsoft SQL Server (RIF.PROD.011).

3.4.6.2 *Storage Service*

Sono i servizi messi a disposizione della logica applicativa che si occupano della memorizzazione di documenti su *filesystem* attraverso repository documentali che consentono di associare metadati al documento. L'Autorità adotta il sistema Alfresco (RIF.PROD.012) come repository documentale asservito ai servizi di business. L'accesso al repository deve essere realizzato utilizzando lo standard Content Management Interoperability Services (CMIS).

3.4.6.3 *Componenti di business intelligence*

L'autorità ha sviluppato un proprio servizio di business intelligence SISk (Sistema conoscitivo appalti pubblici) realizzato con MicroStrategy (RIF.PROD.013), finalizzato a fornire agli utenti interni dell'Autorità indicazioni sui dati sulle gare mediante circa 200 report predefiniti ed uno strumento di navigazione libera con circa 130 indicatori. Le funzionalità offerte, oltre a quella di ricerca approfondita di una gara o di un appalto, sono le seguenti:

1. **Monitoraggio del mercato:** riporta analisi statistiche del mercato degli appalti pubblici in Italia, secondo diverse viste e metriche, fornendo la possibilità di analizzare il numero ed importo gare, lotti perfezionati ed aggiudicati secondo diverse dimensioni di analisi (Tipo Settore, Tipo Appalto, Localizzazione Geografica, etc.);
2. **Vigilanza del mercato:** prevede un set di documenti suddivisi in base alle principali fasi del ciclo di vita di una Gara/Appalto. Le fasi disponibili sono: gara, aggiudicazione, esecuzione, e conclusione;
3. **Flussi finanziari entrate:** prevede un set di documenti suddivisi in base alle tipologie di analisi che è possibile effettuare. Le sezioni disponibili sono: analisi storico e previsionale, analisi e monitoraggio entrate SA, analisi e monitoraggio entrate OE, verifica duplicazioni di versamento e riconciliazione MAV;
4. **Analisi libera:** consente la creazione di report personalizzati a partire dagli oggetti messi a disposizione nell'ambiente di BI.

Come illustrato nella successiva figura, l'architettura del sistema è basata su MicroStrategy, che attinge ai dati provenienti da diverse fonti informative come la BDNCP:

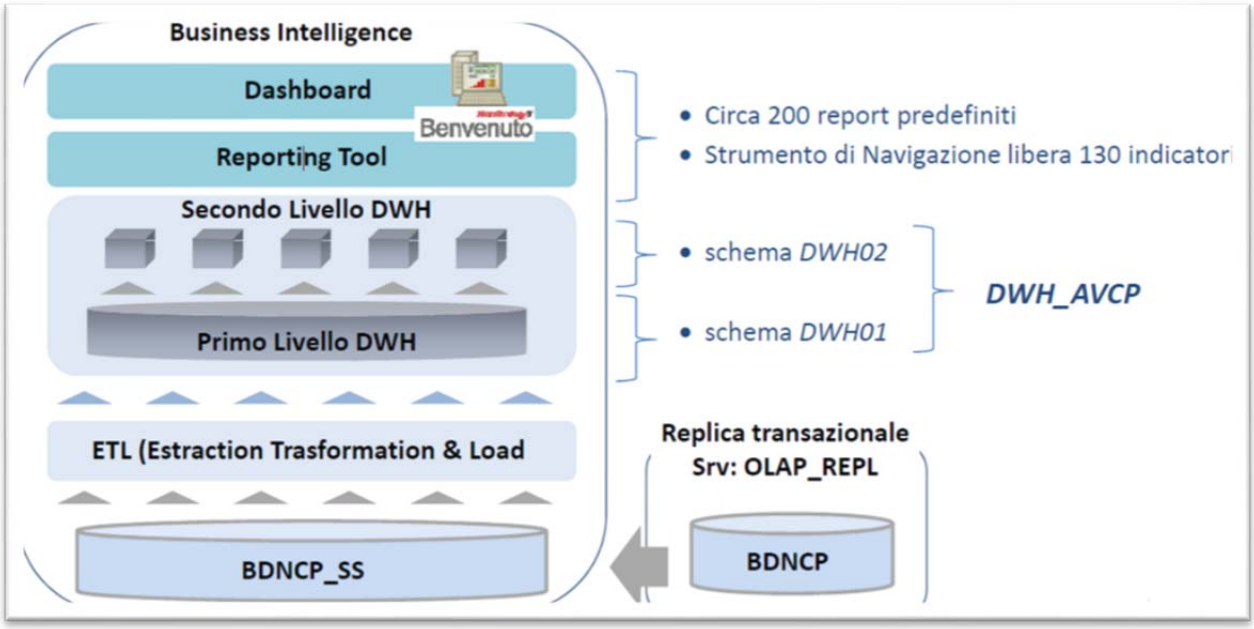

Figura 5 Architettura di Business Intelligence dell'Autorità

3.4.7 Servizi trasversali

Con tali servizi si identificano i componenti che hanno una dimensione trasversale al sistema informativo dell'Autorità, in quanto sono di utilità a tutti i servizi grazie alle funzionalità di base offerte come l'autenticazione, la profilazione degli utenti, la firma digitale, i servizi di logging e auditing, i servizi di messaggistica etc. La centralizzazione di tali funzionalità porta a risparmi sullo sviluppo, sulla manutenzione e sulla gestione da parte dell'Autorità.

3.4.7.1 Sicurezza applicativa (Autenticazione. Autorizzazione)

I servizi ad accesso riservato prevedono che l'utente abbia superato preventivamente la fase di autenticazione e autorizzazione. Per consentire ciò, l'Autorità dispone di una propria infrastruttura di autenticazione e autorizzazione denominata IAM (Identity e Access Management), realizzata tramite PicketLink di Jboss, il quale gestisce, in modo centralizzato, gli accessi alle applicazioni, attraverso un servizio di autenticazione centralizzato LDAP, esposto tramite interfaccia REST o Web Services, per la gestione delle credenziali di accesso degli utenti ,e mediante un servizio centralizzato di gestione dei permessi di accesso alle funzionalità di ogni singola applicazione, esposti tramite interfaccia REST o Web Services. I servizi interni dell'Autorità invece sono acceduti mediante un servizio di autenticazione basato su Active Directory.

3.4.7.2 Service registry e repository

Nell'ottica dell'implementazione di un modello di SOA governance, l'introduzione di un Registro (Registry) consente di sapere quali siano i servizi disponibili e come accedervi. Il Registry è simile ad un indice dove sono registrati e localizzabili i servizi mentre il Repository è un catalogo completo dei servizi disponibili e delle informazioni ad essi correlate che, oltre a consentire di capire cosa fanno e dove sono, aiutano un'organizzazione ad assicurare che vengano utilizzati in maniera appropriata. Attraverso la raccolta e l'archiviazione di tutte le informazioni associate ai servizi (non solo relative all'interfaccia, ma anche alle policy, alle specifiche implementazioni, alle interdipendenze, ecc) attraverso il Repository si promuove un riutilizzo ottimale dei servizi e si migliora il controllo sul deployment e sull'impiego degli stessi, quale parte di una più ampia strategia SOA. In quest'ottica la scelta di un Repository adeguato rende possibile l'individuazione, il governo e la gestione completa del ciclo di vita dei servizi.

3.4.8 Servizi infrastrutturali

Con tali servizi si identificano i componenti dell'infrastruttura dell'Autorità necessari per l'erogazione dei servizi applicativi e dei servizi trasversali. Tra questi si individuano quelli di posta elettronica, i servizi di rete come le VPN, di sicurezza infrastrutturale e perimetrale, i servizi di monitoraggio.

3.5 Ambienti di elaborazione

Per poter gestire correttamente l'intero ciclo di vita del software prodotto e le diverse modalità di erogazione dei servizi sono presenti in Autorità degli ambienti logici, ognuno con un compito ben definito. Di seguito sono descritti tali ambienti con particolare riferimento ai ruoli che ricoprono nel processo di rilascio.

3.5.1 Ambiente di laboratorio

E' un ambiente utilizzato per gli sviluppi interni dell'Autorità, le POC e le sperimentazioni di nuovi prodotti e tecnologie.

3.5.2 Ambiente di rilascio

Viene utilizzato per istallare il software, ottenuto attraverso il processo di build interno dell'Autorità a partire dai sorgenti rilasciati dal fornitore, sul quale vengono eseguite le attività di testing delle funzionalità implementate.

3.5.3 Ambiente di pre-esercizio

Viene utilizzato prima dell'effettivo rilascio in esercizio per consolidare il pacchetto di rilascio, eseguire i test di integrazione e di carico/prestazionali ove necessario e previsto.

3.5.4 Ambiente di esercizio

E' l'ambiente principale dove vengono istallate le componenti software che devono essere esercite, ottenute a valle della fase di realizzazione e il cui esito delle fasi di testing è positivo.

3.5.5 Ambiente di qualificazione

E' un ambiente utilizzato principalmente in ottica di cooperazione applicativa con altri Enti. Viene utilizzato dai consumer per poter verificare e certificare i casi d'uso implementati attraverso i test di integrazione prima dell'abilitazione all'esercizio.

4 Ciclo di vita del software

Nel capitolo vengono descritti i modelli di ciclo di vita del software da utilizzate nell'ambito di ogni singolo intervento realizzato in Autorità.

Verranno descritte, oltre alle metodologie di sviluppo, anche il modello organizzativo adottato e le infrastrutture tecnologiche, che possono variare da sistema a sistema a seconda delle specifiche esigenze e caratteristiche, in modo da avere una visione globale dell'intero processo che il fornitore è tenuto a seguire.

4.1 Fasi progettuali

La definizione di prodotto software, inteso come il software e le informazioni a questo associate, create, modificate o incorporate per soddisfare un obiettivo funzionale nonché contrattuale, include: piani di progetto, requisiti, documenti di analisi funzionale e progettazione, codice sorgente, basi dati, piano dei test, rapporto di esecuzione test, piano di collaudo, manuali utente, manuali d'uso e d'interfaccia dei servizi applicativi, documentazione tecnica di installazione, configurazione e conduzione.

Il Fornitore cura il costante aggiornamento, nel sistema di Configuration Management (CMDB) per le parti di competenza, delle informazioni correlate ad un prodotto software a fronte di un qualunque intervento di sviluppo o manutenzione effettuato.

Di seguito si riportano le fasi del ciclo di vita degli interventi ed i principali prodotti che il fornitore dovrà consegnare per ciascuna fase. Le fasi di seguito elencate possono variare in numero e significato,

ovvero essere diversamente organizzate in cicli per ciascun intervento di sviluppo e manutenzione, concordemente alla metodologia prescelta.

Ogni intervento, indipendentemente dal ciclo di vita scelto, è organizzato nella successione delle fasi di: definizione, analisi, progettazione, realizzazione, accettazione, passaggio in esercizio.

4.1.1 Fase di definizione

La fase di definizione costituisce la prima attività operativa dell'intervento. Viene attivata a seguito della attuazione di un programma annuale o pluriennale di attività o al manifestarsi di una nuova esigenza. In questa fase gli uffici tecnici dell'Autorità definiscono e descrivono formalmente le esigenze della struttura committente destinataria del sistema al fine di trasmetterle al fornitore per le dovute valutazioni.

Soggetti coinvolti:

- Struttura Committente (SC)
- UPSI
- Fornitore

Prodotti di fase attesi:

- Requisiti Utente
- POC
- Piano di lavoro

Attività

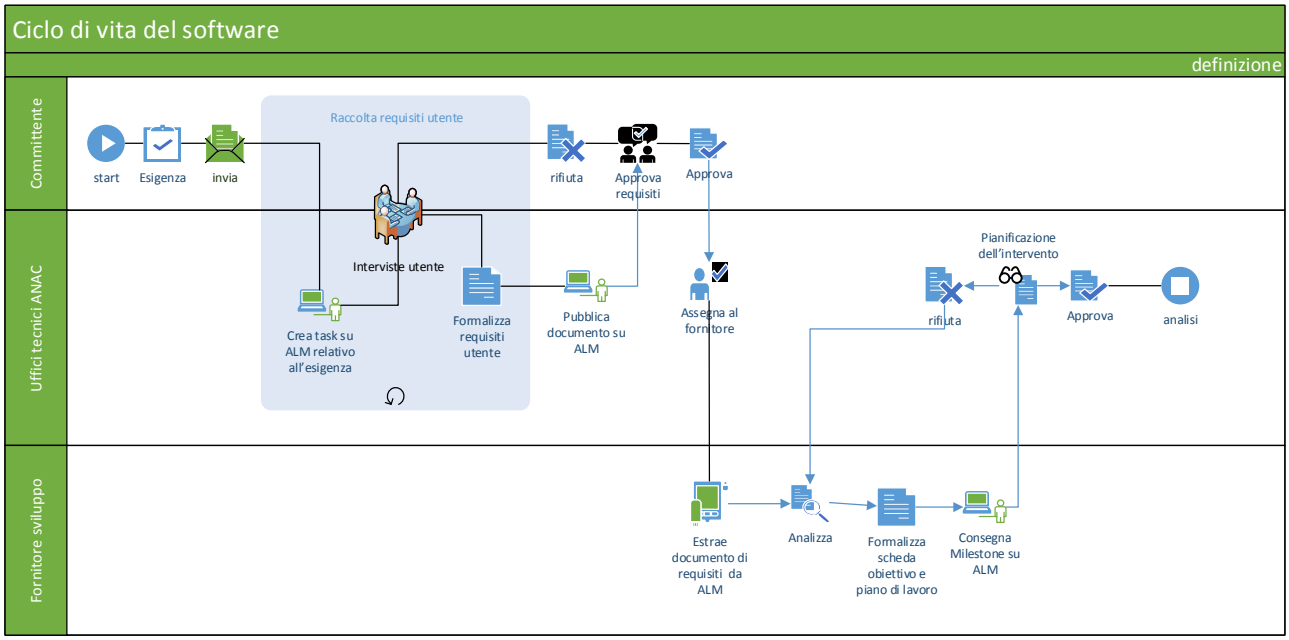

Figura 6 - fase di definizione

A fronte di una nuova esigenza, gli uffici tecnici dell'Autorità effettuano la raccolta dei requisiti e la formalizzazione delle esigenze attraverso gli standard documentali previsti dalle presenti linee guida. All'attività di raccolta e formalizzazione dei fabbisogni partecipa l'ufficio committente che esprime la propria esigenza per mezzo di interviste e documentazione a supporto. Gli uffici dell'Autorità redigono, pertanto, il documento di requisiti utente.

Successivamente l'UPSI codifica il nuovo progetto sulla piattaforma ALM immettendo i singoli requisiti, l'ufficio committente ed il team di progetto che segue le fasi di realizzazione.

Qualora sia ritenuto opportuno, può essere richiesto al fornitore di produrre una POC per valutare aspetti particolari del software richiesto. Tale esigenza si potrebbe verificare in uno dei seguenti casi:

- Il prodotto richiesto è particolarmente innovativo nella veste grafica o nella metafora di interazione con l'utente;
- Il prodotto richiesto fa uso di tecnologie non ancora presenti nei sistemi ANAC;
- Il prodotto richiesto ha un impatto rilevante nell'immagine dell'ANAC nei confronti dei propri *stake holders*;
- Esistono margini di incertezza, da parte dell'ufficio committente, su taluni requisiti funzionali del software da realizzare;

.

L'ufficio UPSImette a disposizione del fornitore il documento di Requisiti Utente sulla piattaforma ALM per la valutazione e redazione del piano di lavoro. Il fornitore consegna il piano di lavoro sulla piattaforma. Il piano di lavoro viene approvato secondo un iter definito dall'Autorità.

4.1.2 Attivazione dell'intervento

L'attivazione consiste nella comunicazione formale al fornitore, da parte dell'UPSI, dell'avvio dell'intervento tramite la consegna della "Nota di avvio". Non sono riconosciute al fornitore attività svolte in assenza o in precedenza alla consegna della nota di avvio ad eccezione di quelle previste nella fase di definizione, ove approvate.

4.1.3 Fase di analisi

Durante la fase di analisi vengono specificate le caratteristiche essenziali della soluzioni da produrre in termini di: funzionalità del software, piano dei test, modello concettuale delle basi dati di riferimento, aspetti di interfaccia utente e dei servizi utilizzati. La definizione del modello della base dati e dei servizi utilizzati dal sistema da sviluppare avviene attraverso la collaborazione tra il fornitore e l'UPSI, in particolare la realizzazione di nuovi servizi a scapito del riuso di quelli esistenti deve essere espressamente portata all'attenzione dell'UPSI e approvata. In questa fase, vengono definiti i casi d'uso ed il piano di test del sistema secondo i seguenti vincoli:

- Ad ogni requisito utente deve corrispondere un caso d'uso (un caso d'uso può riguardare più di un requisito);
- Per ogni caso d'uso debbono essere rappresentati tutti gli scenari prevedibili;
- Ogni scenario di un caso d'uso deve essere interessato in almeno un caso di test;
- Ogni scenario che prevede un interazione con l'utente deve essere rappresentato tramite un mockup grafico dell'interfaccia utente.

Durante l'attività di analisi dei requisiti può essere necessario un approfondimento con il committente per chiarire aspetti dei requisiti utente raccolti nella fase precedente.

Qualora ritenuto necessario ed in funzione del modello di sviluppo prescelto, può essere richiesta la realizzazione di un prototipo che rappresenti almeno le modalità di navigazione e il layout delle interfacce. Per ogni nuovo servizio (WS, REST, EJB) realizzato nell'ambito dell'intervento deve essere rilasciato un mock che ne simuli il comportamento riproducendone l'interfaccia ed una risposta standard.

Soggetti coinvolti:

- Struttura committente

- UPSI
- UESI
- Fornitore

Prodotti di fase attesi:

- Analisi di dettaglio
- Piano di gestione dei rischi
- Piano di test
- Mock-dei servizi da realizzare
- Mock-up interfaccia grafica
- Consolidamento della stima dell'intervento

Attività

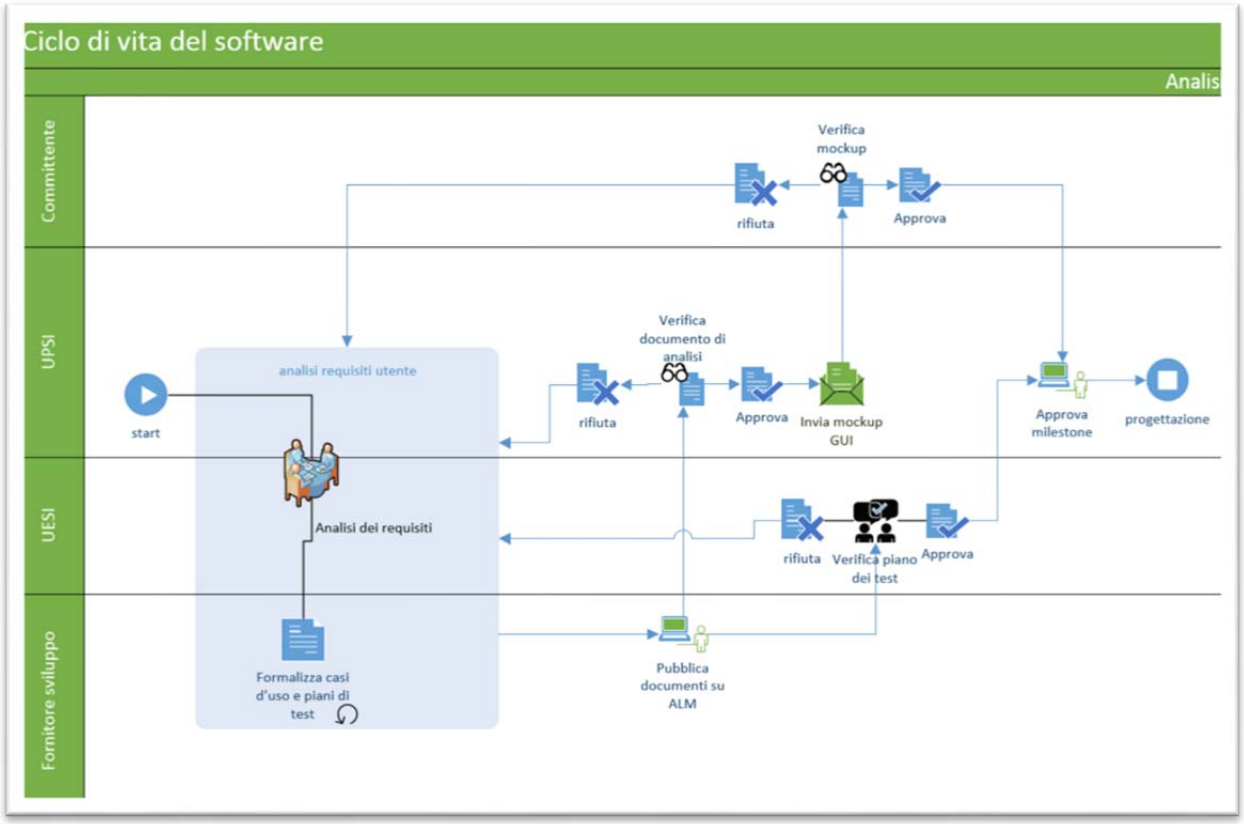

Figura 7 - fase di analisi

Il fornitore elabora i prodotti di fase attesi con la collaborazione dell'ufficio UPSI. L'ufficio UESI viene coinvolto per la definizione del piano di test di carico e integrazione e la sua approvazione. Inoltre, l'ufficio UESI descritte le effettive release attive negli ambienti e gli eventuali piano di evoluzione degli stessi. Nell'eventualità che la realizzazione dell'intervento richieda l'adozione di tecnologie o prodotti software di terze parti non ancora installati presso la server farm dell'Autorità, l'UESI viene coinvolto nella definizione del documento "Analisi di dettaglio" ed approva espressamente il prodotto. Tutti i prodotti di fase vengono ricevuti ed approvati da UPSI. In caso di mancata approvazione di uno o più prodotti di fase il fornitore apporta le necessarie correzioni alla documentazione secondo quanto specificato dall'ufficio UPSI.

4.1.4 Fase di progettazione

La fase di progettazione è volta a tradurre tutte le caratteristiche della soluzione in specifiche tecniche di dettaglio necessarie alla realizzazione del software. Viene definita l'architettura della soluzione completa della specifica dei componenti da realizzare. La definizione della fase esecutiva può rendere necessaria

la revisione di uno o più prodotti delle attività delle fasi precedenti. In questa fase si ha il consolidamento e la stesura definitiva del piano di test.

Soggetti coinvolti:

- UPSI
- UESI
- Fornitore

Prodotti di fase attesi:

- Specifiche di Intervento
- Documento di progettazione dei servizi
- Documento di interfaccia dei servizi
- Piano di test definitivo
- Indicatori di qualità della fase

Attività

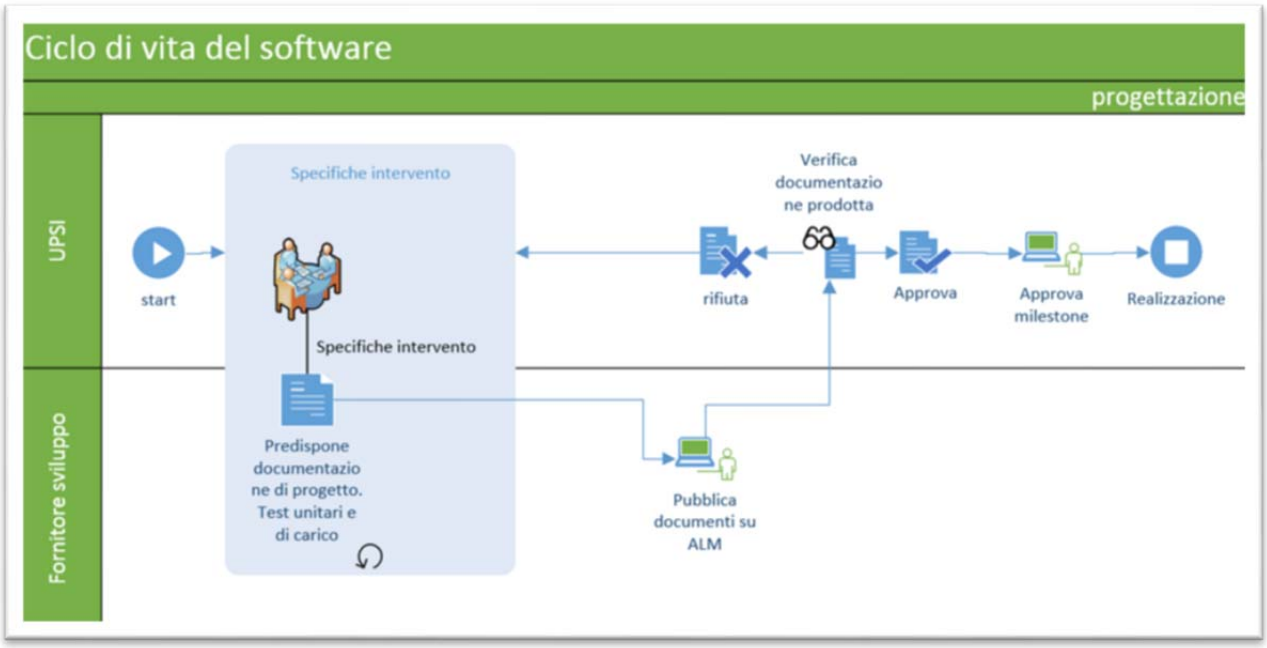

Figura 8 - fase di progettazione

Il fornitore elabora i prodotti di fase attesi. Tutti i prodotti di fase vengono ricevuti ed approvati da UPSI previa approvazione del piano dei test da parte dell'ufficio UESI. In caso di mancata

approvazione di uno o più prodotti di fase il fornitore apporta le necessarie correzioni alla documentazione secondo quanto specificato dall'ufficio UPSI.

4.1.5 Fase di realizzazione

La fase di realizzazione ha lo scopo di produrre le componenti software e la base dati che costituiscono il sistema. Il software prodotto deve essere accompagnato da adeguata documentazione che ne illustri le modalità di installazione, configurazione, gestione ed uso; nonché da tutti gli strumenti che consentano di verificare la correttezza di quanto prodotto rispetto alle attese.

Soggetti coinvolti:

- Fornitore

Prodotti di fase attesi:

- Codice sorgente
- Documentazione dei sorgenti generata automaticamente (ove applicabile)
- Codice a corredo del sorgente
- Lista oggetti software
- Manuale di installazione e gestione
- Manuale utente
- Codice per l'esecuzione dei test unitari
- Codice per il test delle interfacce GUI
- Certificazione ed esito dei test eseguiti

Attività

Il fornitore elabora i prodotti di fase attesi e li consegna sulla piattaforma ALM. L'approvazione dei prodotti software è della documentazione è posticipata alla successiva fase di accettazione.

4.1.6 Accettazione del software

La fase di accettazione ha lo scopo di verificare la qualità e la correttezza rispetto agli standard qualitativi e ai requisiti funzionali del software prodotto.

Soggetti coinvolti:

- UPSI

- UESI
- Fornitore

Prodotti di fase attesi:

- Approvazione della consegna

Attività

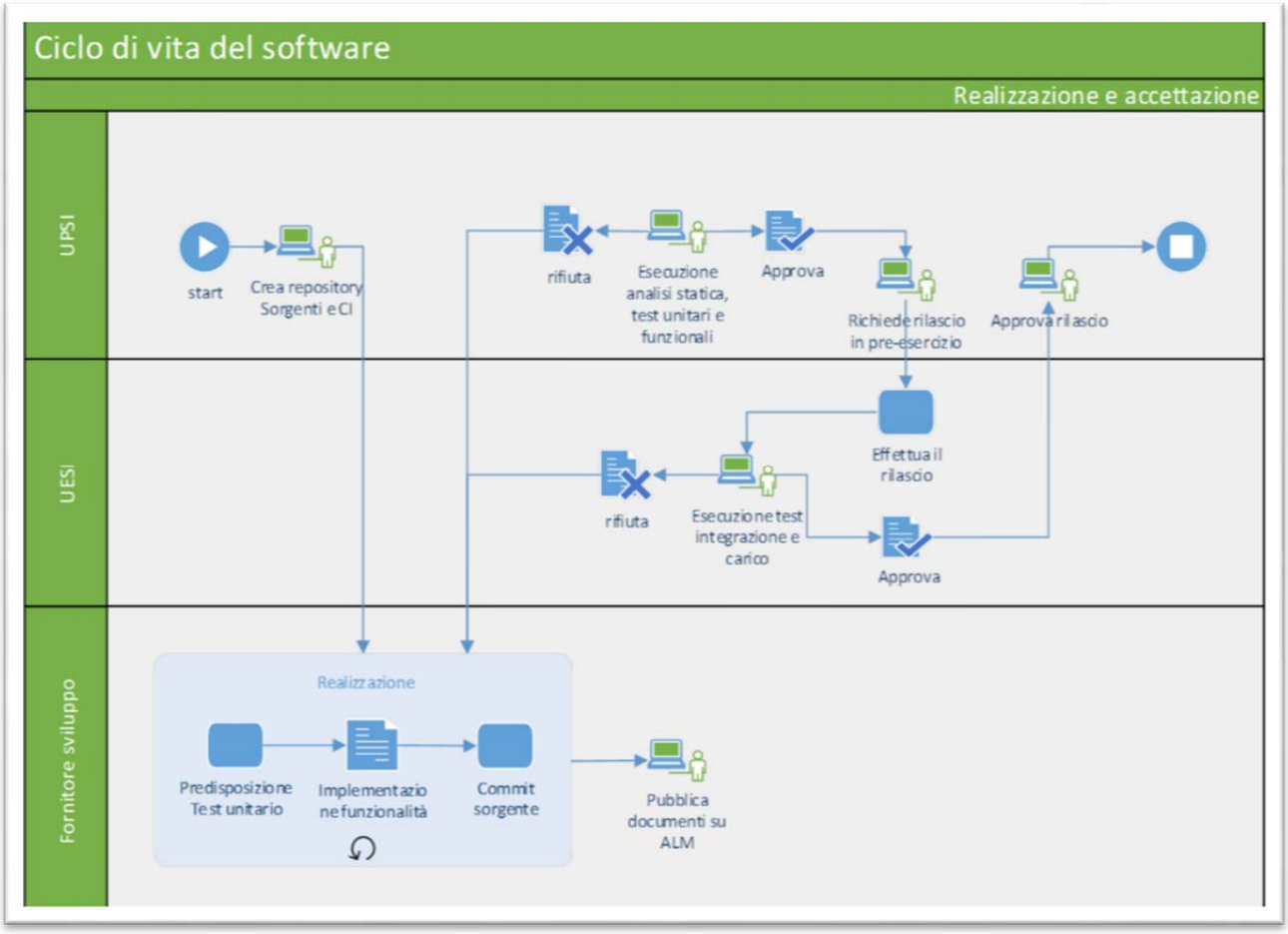

Figura 9 - fase di realizzazione e accettazione

L'UPSI tramite la componente di CI presente nella piattaforma ALM provvede ad eseguire:

- un analisi statica del codice verificandone la qualità e la correttezza secondo gli standard ISO di riferimento (ISO/IEC 25010 (ISO/IEC 9126))
- La compilazione del codice sorgente rilasciato attraverso script MAVEN
- L'esecuzione dei test unitari JUNIT rilasciati congiuntamente al codice sorgente
- Il rilascio sul repository degli artefatti dei pacchetti prodotti seguendo le regole di versionamento degli stessi.

- L'installazione dei pacchetti attraverso script automatici (ove possibile) in ambiente di rilascio.
- L'esecuzione dei test funzionali automatizzati.

In caso di anomalie riscontrate il sistema apre in automatico un ticket sulla piattaforma ALM che viene assegnato al fornitore.

In caso di successo di tutti i passi sopra elencati i pacchetti software vengono installati e configurati in ambiente di Pre-Esercizio dall'ufficio UESI che procede poi all'esecuzione dei test di integrazione e, ove ritenuto necessario, ai test di carico del software.

Ad esito positivo l'UPSI notifica al fornitore l'accettazione dei prodotti di fase.

4.1.7 Collaudo ed esercizio

La disciplina della fase di collaudo e del successivo rilascio in esercizio sono escluse dal campo di applicazione del presente documento.

4.1.8 Criterio di chiusura delle fasi

La fine di ogni fase richiede la consegna dei documenti e dei prodotti di fase previsti. Si sottolinea che l'avvenuta consegna dei prodotti di fase non ne implica l'accettazione e non esclude la possibilità di dover apportare modifiche a fronte delle verifiche effettuate.

4.2 Forma e contenuto dei prodotti di fase

4.2.1 Documento Requisiti Utente (RU)

Il documento Requisiti Utente esprime le esigenze del soggetto committente formalizzandole in singoli requisiti funzionali e non funzionali che descrivono il sistema richiesto nella sua interezza. Ogni requisito deve esprimere concetti semplici e non ulteriormente scomponibili ed essere formulato in proposizioni assertive di forma sia positiva, sia negativa, evitando il tempo futuro dei verbi.

Ciascun requisito è individuato da un identificativo univoco nella forma [(RF|RNF)-AM-AR.nn], dove:

- RF Requisito Funzionale;
- RNF Requisito Non Funzionale;
- AM identifica l'ambito;
- AR identifica l'area funzionale;
- nn è un progressivo numerico all'interno dell'area funzionale.

Per ogni requisito è previsto un codice di versionamento ed uno stato. Lo stato può assumerei i valori:

- In definizione: il requisito non è ancora approvato

- Approvato: il requisito impegna il fornitore e deve essere realizzato
- Sostituito: il requisito è stato sostituito da una nuova versione approvata
- Annullato: il requisito non è più in vigore

Il documento è organizzato nelle seguenti sezioni:

- Requisiti Funzionali: descrive le funzionalità che il sistema richiesto deve erogare agli utenti
- Requisiti di accesso: definisce i ruoli utente del sistema ed attribuisce a ciascun ruolo le sue funzionalità
- Requisiti di Sicurezza: definisce i requisiti di identificazione e autorizzazione degli utenti del sistema
- RNF

4.2.2 Proof Of Concept (POC)

Le Proof of Concept consistono nella realizzazione parziale di un componente o di un intero progetto finalizzato a dimostrarne la fattibilità o la fondatezza. Per la realizzazione di una POC il fornitore deve consegnare il codice sorgente e il codice correlato nonché il manuale di installazione e configurazione. Di norma la realizzazione di una POC prevede la predisposizione di un ambiente ad hoc presso la server farm dell'Autorità.

4.2.3 Piano di Lavoro

Il Piano di Lavoro (PdL) costituisce lo strumento di supporto alla gestione complessiva delle attività previste per un intervento. Il documento si articola in due sezioni:

- Descrizione dell'Intervento: in cui vengono declinati obiettivi, ambito ed approccio progettuale
- Piano delle attività, in cui viene presentato il piano delle attività con declinazione di tempi e costi dell'intervento

Nella rappresentazione del piano delle attività è richiesta la produzione del diagramma Gantt del progetto nonché la definizione della WBS e dei singoli work package.

4.2.4 Analisi di dettaglio

Il documento di Analisi di dettaglio da evidenza e formalizza le caratteristiche essenziali e il modello di funzionamento del software che si andrà a sviluppare. La descrizione formale del prodotto viene realizzata utilizzando le seguenti convenzioni o linguaggi:

- Diagrammi dei casi d'uso
- Modello concettuale dei dati E-R

- Elenco delle nuove funzioni e delle funzioni condivise
- Matrice delle funzionalità
- Elenco dei servizi oggetto di riuso
- Elenco dei servizi da realizzare

4.2.5 Piano di test

Il documento descrive la strategia e casi di test per il collaudo del sistema in sviluppo. I casi di test sono organizzati in cicli, ciascun ciclo di test è caratterizzato da criteri di ingresso e di uscita: i criteri di ingresso sono rappresentati dalle condizioni e dai requisiti per i quali il ciclo di test può avere inizio; i criteri di uscita sono rappresentati dalle condizioni per le quali termina la sessione di test. I criteri di ingresso e di uscita dai cicli di test costituiscono uno dei parametri che determinano, attraverso le regole di propedeuticità in essi definite, la sequenza con cui vengono eseguiti i test.

I test previsti sono:

- Test unitari
- Test funzionali
- Test d'integrazione
- Test di carico

4.2.6 Piano di gestione dei rischi per l'intervento

Descrive la modalità di gestione dei rischi nell'erogazione di tutti i servizi contrattualmente previsti; il piano tiene traccia di tutti i rischi identificati, delle valutazioni di impatto e probabilità, delle contromisure previste e dello stato di applicazione.

Nel corso della realizzazione il fornitore dovrà tenere aggiornato l'elenco dei rischi individuati per ogni servizio o intervento con le azioni di riduzione associate ed il relativo stato di attuazione.

Le azioni di riduzione saranno discusse ed approvate nel corso delle riunioni di stato avanzamento lavori.

4.2.7 Mockup

Il mockup costituisce un modello non funzionante del prodotto che permetta di valutare e di apportare eventuali modifiche all'aspetto finale. In particolare per le interfacce utente è richiesta la produzione di una o più pagine web in numero sufficiente a consentire di valutare l'usabilità, la metafora di interazione ed in generale l'esperienza utente del prodotto da parte degli utenti finali. Per i servizi che verranno

modificati o prodotti ex novo nell'intervento è richiesta la produzione di un modello che ne riproduca l'interfaccia. L'interrogazione del servizio mock restituisce dei dati predefiniti.
Template di riferimento: non applicabile.

4.2.8 Specifiche di Intervento

Costituisce il documento di progettazione esecutiva che deve essere seguito per la realizzazione del software. Le soluzioni progettuali sono di norma rappresentate secondo lo standard UML. La struttura interna del sistema viene rappresentata attraverso uno o più component diagram interconnessi. Il prodotto viene rappresentato attraverso i seguenti modelli.
Modello statico:

- class diagram
- package diagram
- Diagramma di contesto
- Component diagram

Modello dinamico:

- Activity diagram
- Sequence diagram

Base dati

- Schema concettuale logico e fisico
- Elenco delle entità aggiunte o aggiornate
- Tracciato delle tabelle aggiunte o aggiornate

4.2.9 Documento di progettazione dei servizi

Il documento descrive il comportamento ed il funzionamento di ciascun servizio interessato dall'intervento mediante:

- Sequence diagram
- Activity diagram
- Collaboration diagram
- State diagram
- Copertura dei casi d'uso

4.2.10 Documento di interfaccia dei servizi

Il documento descrive l'interfaccia di di ciascun servizio interessato dall'intervento mediante:

- Descrizione del flusso d'interoperabilità
- Interfaccia EJB
- Interfaccia Web Service
- Interfaccia RestFull
- Struttura del messaggio di ritorno
- Codici di ritorno
- Codici di errore

4.2.11 Codice sorgente

Il codice software sviluppato dal fornitore per la realizzazione del prodotto richiesto. La consegna avviene sul sistema ALM secondo le specifiche operative previste.

4.2.12 Documentazione del software generata automaticamente

Il codice sorgente deve essere adeguatamente commentato, in modo da consentire la generazione in via automatica della documentazione del software consegnata in allegato al codice sorgente.

4.2.13 Codice a corredo del sorgente

È costituito dell'insieme del codice necessario alla corretta installazione e funzionamento del codice sorgente prodotto. Il codice a corredo può essere prodotto da terze parti e deve essere liberamente reperibile sul mercato, può essere consegnato in forma di codice sorgente o compilato. Se il codice a corredo è tutelato o sottoposto a licenze d'uso il fornito si impegna a cedere la licenza d'uso ad ANAC.

4.2.14 Lista oggetti software

La lista oggetti software contiene l'elenco dei singoli moduli software, anche di terze parti, realizzati o aggiornati nel corso dell'intervento. Per ogni oggetto deve essere indicato il numero di versione precedente e quello assegnato a seguito dell'aggiornamento.
Sono considerati oggetti software:

- File di tipo jar, war, ear
- Librerie software
- File di configurazione
- Store Procedure e Function di Database
- DTSX per gli ETL
- Script di aggiornamento del Database

- File di configurazione datasource
- Template di interfaccia web o documenti
- File di tipo xsd, xslt
- File di tipo wsdl
- Script di shell

4.2.15 Manuale di installazione e gestione

Il manuale di installazione e gestione descrive tutte le procedure necessarie al rilascio e alla gestione della componete applicativa. In particolare descrive:

- L'architettura di massima
- la configurazione e il deployment degli item sui layer dell'architettura
- I contenuti del setup
- Le procedure di installazione aggiornamento e disinstallazione
- La procedura di gestione

Il documento viene validato dall'ufficio UESI in quanto principale fruitore dello stesso. L'ufficio UESI a partire dal documento predisporrà gli script necessari per l'automazione del processo di rilascio nei vari ambienti

4.2.16 Manuale utente

Il manuale utente descrive le funzionalità implementate dal servizio/sistema e mette in condizione l'utente finale di utilizzare l'applicazione. Il documento deve:

- Descrivere le modalità per la fruizione/accesso al servizio/sistema indicando tutte le pre-condizioni necessarie per accedere allo stesso (sia tecnologiche che funzionali) che devono essere dettagliate nei paragrafi successivi
- Descrivere la procedura di registrazione per l'accesso al servizio/sistema
- Descrivere come è possibile accedere alle varie funzionalità attraverso i menù di navigazione e gli strumenti di ausilio al servizio
- Descrivere la funzionalità.
- Descrivere la pagina ottenuta a valle di una comunicazione.

4.2.17 Codice per l'esecuzione dei test unitari

Il pacchetto è normalmente incluso nello sviluppo e fa parte del codice sorgente dell'intervento. Il codice per i test unitari predispone le precondizioni all'esecuzione delle singole unità di software realizzato, le esegue e verifica se i risultati ottenuti sono conformi alle specifiche.

4.2.18 Codice per il test delle interfacce web

Il pacchetto contiene il software per la configurazione e per la predisposizione dell'ambiente di test delle interfaccie web. L'Autorità utilizza il pacchetto Selenium per l'esecuzione del test.

4.2.19 Certificazione dei test eseguiti

Il fornitore restituisce il documento Piano dei test ove è debitamente annotata e certificata l'esecuzione dei singoli test nell'ambiente di sviluppo situato presso il fornitore

4.2.20 Consolidamento della dimensione del software

Ad esito della conclusione della fase di analisi il fornitore propone una misurazione accurata del volume dell'intervento, ad aggiornamento della stima presentata nel Piano di lavoro

4.3 Tipologie di cicli di vita

Di seguito vengono descritti i tipi di cicli di vita del software oggi in uso in Autorità. La scelta del tipo di ciclo di vita da adottare è demandata all'Autorità nel momento dell'attivazione dell'obiettivo.

4.3.1 Ciclo completo

Il ciclo completo di sviluppo prevede l'adozione di un modello a cascata suddiviso nelle fasi di: definizione, analisi, progettazione, realizzazione, accettazione. Al termine di ciascuna fase è prevista una attività di valutazione dei prodotti realizzati. L'approvazione degli stessi consente il passaggio alla fase successiva. I singoli stadi possono prevedere cicli interni sia per la realizzazione dei prodotti di fase che per componenti prototipali.

Il ciclo completo viene adottato nel caso di interventi per i quali sia possibile definire dei requisiti utente ragionevolmente stabili e comunque nei casi seguenti:

- Sviluppo di nuove applicazioni per le quali è disponibile una analisi preliminare o uno studio di fattibilità
- Manutenzioni evolutive che coinvolgono una parte consistente dell'applicazione

4.3.2 Ciclo ridotto

Il ciclo ridotto costituisce una semplificazione del ciclo completo ed adotta il medesimo modello a cascata con la riduzione dei prodotti di fase richiesti al fornitore.

Il ciclo ridotto viene adottato nel caso di interventi di manutenzione evolutiva, correttiva e adeguativa per i quali sia possibile definire dei requisiti utente ragionevolmente stabili.

4.3.3 Ciclo unico

Il ciclo unico viene utilizzato per la realizzazione di interventi di manutenzione evolutiva, correttiva e adeguativa con carattere di urgenza o di particolare semplicità. Nel ciclo unico le fasi di Analilisi e Progettazione coincidono e possono essere accorpate anche alla fase di Realizzazione.

4.3.4 Ciclo a spirale

Per particolari ed eccezionali esigenze, legate alla sperimentazione preliminare all'introduzione nella server farm dell'Autorità di nuovi prodotti o tecnologie; per la realizzazione di prodotti ad alto contenuto di innovazione o per interventi di notevoli complessità ove non sia possibile definire requisiti utente stabili, è possibile adottare un ciclo di produzione a spirale. In questo caso la successione delle fasi di Definizione, Analisi, Progettazione e Realizzazione, può essere ripetuta più volte, fino a un massimo definito dall'Autorità nella fase di attivazione. I prodotti di fase sono quelli previsti dal ciclo ridotto.

Nella fase di attivazione l'Autorità definisce:

- Il numero massimo di cicli previsti per l'intervento
- Le eventuali modalità di interruzione dell'intervento prima del raggiungimento dell'obiettivo

4.3.5 Riepilogo

La tabella seguente descrive in via sintetica i cicli di vita.

Legenda :

- **Fase:** contiene le fasi in cui è scomposto il ciclo di vita;
- **Prodotti di fase:** contiene i prodotti di output della singola fase;
- **Ciclo di vita:** definisce par ciascuno dei tre cicli di vita individuati i prodotti di fase applicabili secondo la codifica seguente:
  - **M** – Mandatorio
  - **O** - Opzionale
  - **P** – Posticipato ad una fase successiva

- **X** – Non previsto
- **A** – soggetto ad esplicita Approvazione per il passaggio di fase

- **Responsabilità:** definisce la matrice di assegnazione responsabilità per le attività o i prodotti di fase secondo la seguente codifica:
  - **R – Responsible,** Responsabile dell'esecuzione dell'attività - realizzazione del prodotto
  - **A – Accountable**, Approva il prodotto o l'attività
  - **C – Consulted**, Collabora all'esecuzione dell'attività o alla realizzazione del prodotto
  - **I – Informed,** Informato dell'attività
  - **X** – non coinvolto

Fase	Prodotti di fase	Ciclo di vita		
		Completo	Ridotto	Unico
Definizione	Documento Requisiti Utente	M	M	M
	POC (opzionale)	O	O	X
	Piano di lavoro obiettivo	MA	MA	MA
Attivazione	Nota di avvio	M	M	M
Analisi	Analisi di dettaglio dei requisiti funzionali e non funzionali	MA	MA	MA
	Piano di gestione dei rischi	MA	MA	OA
	Mock dei servizi da realizzare	M	O	X
	Mock up interfaccia grafica	M	O	X
	Prototipo	O	O	O
	Piano di test	MA	MA	MA
	Consolidamento della	MA	O	X

	stima dell'intervento			
Progettazione	Architettura della soluzione, Modello logico e fisico dei dati	MA	OA	X
	Modello statico e dinamico delle componenti	MA	OA	X
	Progetto dei servizi e delle interfacce	M	M	M
	Consolidamento del piano di test	M	F	X
Realizzazione	Codice sorgente	M	M	M
	Codice test unitari	M	M	M
	Codice di test GUI	M	M	X
	Software a corredo del sorgente	F	F	F
	Manuale di installazione e gestione del software	M	M	M
	Documentazione utente	M	M	M
	Certificazione ed esito dei test eseguiti	M	M	M
	Lista oggetti software	M	F	F
Accettazione del software	Approvazione della consegna			

4.4 Metodologie e prodotti utilizzati

Per gestire l'intero ciclo di vita del progetto l'Autorità utilizza una piattaforma Application Lifecycle Management (ALM), costituita da un insieme di software opensource, finalizzata a gestire ed automatizzare l'intero processo di sviluppo e di rilascio in ottica Devops.

La piattaforma ALM rappresenta l'unione delle attività di gestione con le attività di ingegneria del software, resa possibile dall'utilizzo di strumenti che facilitano la gestione delle fasi di: analisi dei requisiti, progetto architetturale, sviluppo, testing, gestione delle release, del change e del deployment.

Un processo efficace di Application Lifecycle Management, supportato da strumenti adeguati può:

- Aumentare la produttività, poiché il gruppo di lavoro condivide le pratiche di sviluppo e deployment, e gli sviluppatori possono focalizzarsi sui requisiti di business correnti.
- Migliorare la qualità, in modo che il prodotto applicativo soddisfi i bisogni e le aspettative degli utenti
- Eliminare le barriere di comunicazione attraverso l'efficace collaborazione ed il facile flusso di informazioni
- Accelerare lo sviluppo semplificando l'integrazione
- Tagliare i tempi di manutenzione sincronizzando lo sviluppo e la progettazione
- Massimizzare l'investimento sulle capacità, sui processi e sulle tecnologie
- Aumentare la flessibilità, riducendo il tempo ed i costi di integrazione con nuove applicazioni che soddisfano nuovi requisiti di business

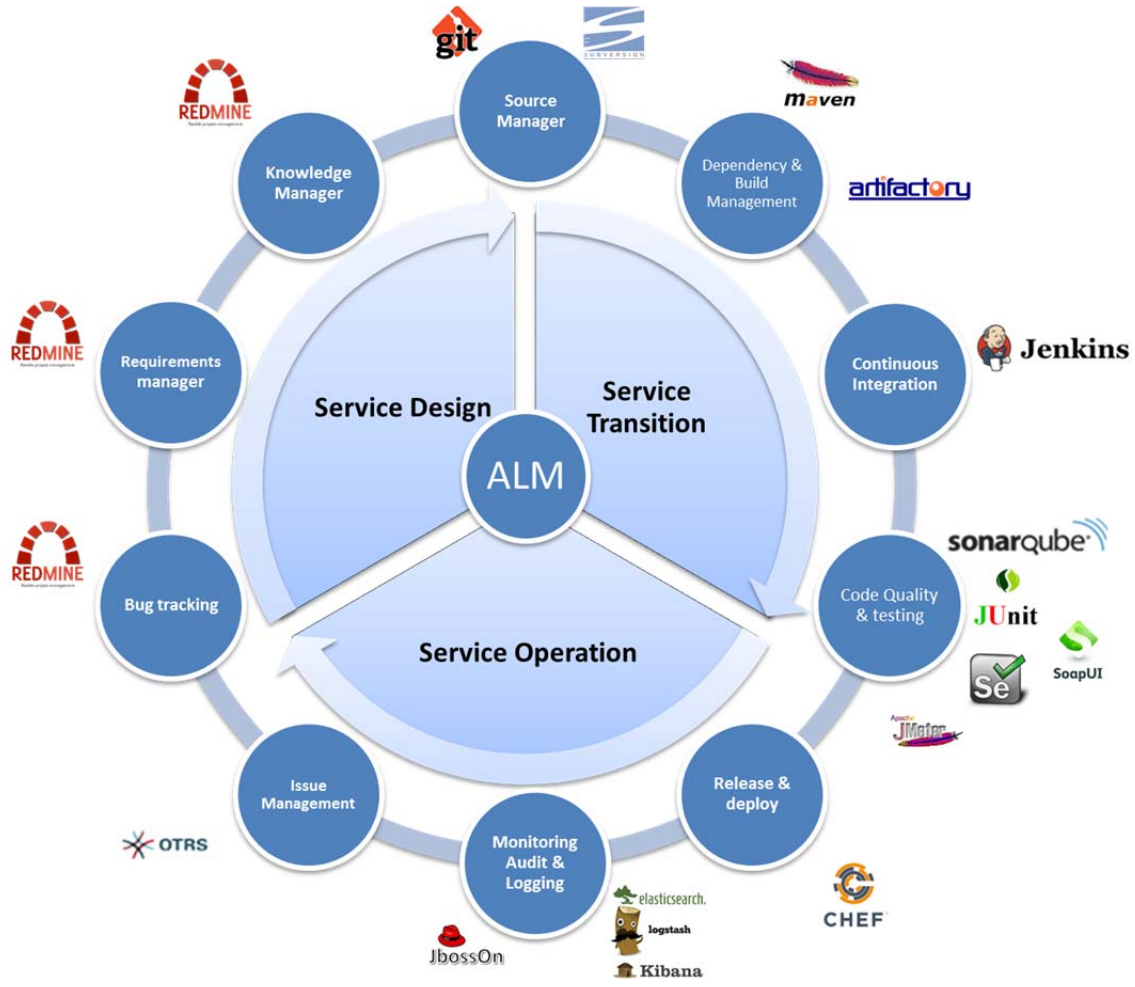

Figura 10 - Processo ALM e software di supporto

Di seguito verranno descritte le singole componenti della piattaforma adottata da ANAC e le metodologie utilizzate.

4.4.1 Requirement Manager

Il sistema di requirement management, integrato nella piattaforma Redmine, consente l'inserimento dei requisiti utente come issue di progetto. Su ogni item il committente ed il team di progetto dell'Autorità hanno la possibilità di aggiungere commenti e aprire una discussione finalizzata al consolidamento del requisito stesso.

4.4.2 Knowledge Manager

Il sistema di Knowledge management ha come finalità la gestione e la condivisione della conoscenza attraverso un insieme di strategie e metodi per identificare, raccogliere, sviluppare, conservare e rendere accessibile la conoscenza delle persone che fanno parte di una organizzazione.
Attraverso la piattaforma Redmine l'Autorità per ogni singolo progetto raccoglie la documentazione ne gestisce un wiki ed un forum.

4.4.3 Source Manager

Il controllo di versione del codice è un sistema che registra, nel tempo, i cambiamenti ad un file o ad una serie di file, così da poter richiamare una specifica versione in un secondo momento. Un VCS consente allo sviluppatore di ripristinare i file ad una versione precedente, ripristinare l'intero progetto a uno stato precedente, revisionare le modifiche fatte nel tempo, vedere chi ha cambiato qualcosa che può aver causato un problema, l'Autorità dispone nella sua piattaforma ALM dei seguenti sistemi di Versionamento:

- SVN: Subversion Server basato sulla piattaforma Subversion Edge di Collabnet
- GIT: Repository GIT basato su GitLab Comunity Edition

Tali strumenti dovranno essere utilizzati per tracciare tutte le versioni consolidate e candidate all'installazione dei vari ambienti. Per la gestione del versionamento alle linee guida di sviluppo del software

4.4.4 Dependency & build Management

Attraverso la piattaforma è possibile:

- Gestire i servizi: progetti e attività possono essere facilmente inserite, assegnate alle risorse e monitorate.
- Gestire le risorse: comunicare ed assegnare i lavori alle risorse, sapere sempre chi fa cosa e quando.
- Gestire budget e costi: ogni risorsa che viene impiegata su una commessa apporta il proprio costo. Osserva lo stato di salute delle tue commesse in tempo reale.

L'Autorità dispone di una piattaforma opensource basata su Redmine. Nella piattaforma dovranno essere inseriti e gestiti:

- I requisiti del progetto raccolti e consolidati dall'Autorità
- La documentazione prodotta
- il wiki di progetto
- Forum di progetto
- Le versioni del servizio rilasciate ed in corso di rilascio con il riferimento ai requisiti presenti ed alle eventuali correttive.
- il collegamento al repository dei sorgenti

4.4.5 Support ticket

Il sistema gestisce, registra e traccia le richieste di assistenza o di problemi. I sistemi di registrazione delle richieste vengono usati spesso dai centri di assistenza telefonica, per registrare, aggiornare e risolvere i problemi segnalati dai clienti o dagli stessi impiegati dell'azienda. Rientra nel novero degli strumenti utilizzati per interagire con il cliente.

Per questa funzionalità l'Autorità utilizza la piattaforma OTRS.

4.4.6 Bug tracking

Un sistema di bug tracking è un applicativo che consente di tracciare le segnalazioni di bug all'interno dei software, in modo che tali errori siano mantenuti sotto controllo, con una descrizione della riproducibilità e dei dettagli ad essi correlati, e dunque più facilmente risolvibili.

A tal fine l'autorità utilizza le funzionalità disponibile nel prodotto Redmine

4.4.7 Source Control

Il controllo di versione del codice è un sistema che registra, nel tempo, i cambiamenti ad un file o ad una serie di file, così da poter richiamare una specifica versione in un secondo momento. Un VCS consente allo sviluppatore di ripristinare i file ad una versione precedente, ripristinare l'intero progetto a

uno stato precedente, revisionare le modifiche fatte nel tempo, vedere chi ha cambiato qualcosa che può aver causato un problema, l'Autorità dispone nella sua piattaforma ALM dei seguenti sistemi di Versionamento:

- SVN: Subversion Server basato sulla piattaforma Subversion Edge di Collabnet
- GIT: Repository GIT basato su GitLab Comunity Edition

Tali strumenti dovranno essere utilizzati per tracciare tutte le versioni consolidate e candidate all'installazione dei vari ambienti. Per la gestione del versionamento fare riferimento alle linee guida sviluppo software

4.4.8 Sistema di build automation e Testing

L'Automazione dello sviluppo è l'atto di scrivere o automatizzare un'ampia varietà di compiti che gli sviluppatori software fanno nelle loro attività quotidiane di sviluppo come:

- Compilazione del codice sorgente in codice binario;
- Gestione delle dipendenze
- Pacchettizzazione del codice binario;
- Esecuzione di test unitari;
- Deployment di sistemi di produzione;
- Creazione di documentazione e/o note di rilascio.

Ogni progetto java deve utilizzare il framewotk MAVEN per *il build automation* dei progetti e per la gestione delle dipendenze. Maven usa un costrutto conosciuto come Project Object Model (POM); un file XML che descrive le dipendenze fra il progetto e le varie versioni di librerie necessarie nonché le dipendenze fra di esse. In questo modo si separano le librerie dalla directory di progetto utilizzando questo file descrittivo per definirne le relazioni.

Maven effettua automaticamente il download di librerie Java e plug-in Maven dai vari repository definiti scaricandoli in locale o in un repository centralizzato lato sviluppo. L'Autorità dispone di un repository interno basato su Artifactory dove sono memorizzate tutte le librerie presenti in produzione. Tramite MAVEN è possibile eseguire i test unitari realizzati a copertura dei casi d'uso previsti nel progetto.

4.4.9 Sistema di continuous integration

L'integrazione continua (continuous integration in inglese, spesso abbreviato in CI) è una pratica che si applica in contesti in cui lo sviluppo del software avviene attraverso un sistema di versioning. Consiste nell'allineamento frequente dagli ambienti di lavoro degli sviluppatori verso l'ambiente condiviso (mainline).

L'autorità utilizza Jenkins come piattaforma di CI integrata con i repository di codice sorgente SVN e GIT e con il repository di artefatti Artifactory. Ad ogni Commit eseguito sulla piattaforma VCN il sistema di CI esegue l'analisi statica del codice la build tramite Maven ed i test unitari (JUnit).

4.4.10 Sistema di continuous deployment

I sistemi di continuous deployment nascono come estensione del processo di continuous integration con l'obbiettivo di minimizzare il tempo di rilascio in produzione di una nuova versione del software.
Per raggiungere l'obbiettivo atteso il team di progetto utilizza un'infrastruttura che automatizza i vari passaggi che portano al rilascio in produzione.
Per questa funzionalità l'Autorità utilizza la piattaforma Chef.

4.4.11 Sistema di Knowledge management

Il sistema di Knowledge management ha come finalità la gestione e la condivisione della conoscenza attraverso un insieme di strategie e metodi per identificare, raccogliere, sviluppare, conservare e rendere accessibile la conoscenza delle persone che fanno parte di una organizzazione.
A tal fine l'autorità utilizza le funzionalità disponibile nel prodotto Redmine

4.4.12 Sistema di misurazione della qualità del software

Il sistema consente di verificare in maniera automatica la qualità del software rilasciato dal fornitore sul repository secondo gli standard (**RIF.EXT.002)**
La piattaforma utilizzata dell'Autorità è Sonar QUBE.

4.5 Processo di manutenzione

4.5.1 Manutenzione evolutiva

La Manutenzione Evolutiva (MEV) si riferisce all'introduzione di nuove funzioni, o la modifica di funzioni preesistenti, sul parco applicativo dell'Autorità volte a soddisfare esigenze utente che riguardano funzioni aggiuntive, modificate o complementari al sistema esistente. Il servizio comprende pertanto tutti gli interventi finalizzati a migliorare le applicazioni attraverso lo sviluppo di nuove funzionalità, la modifica di funzionalità esistenti, l'integrazione con altre funzionalità/servizi.
Per la realizzazione di interventi di Manutenzione Evolutiva possono essere utilizzati i cicli di vita Completo o Ridotto, in funzione della complessità o consistenza delle nuove funzionalità.
L'esecuzione di un ciclo di Manutenzione Evolutiva prevede l'aggiornamento ed il conseguente versionamento di tutti i documenti previsti come prodotto di fase.

4.5.2 Manutenzione correttiva

Per Manutenzione Correttiva (MAC) si intendono tutte le attività di analisi, diagnosi e rimozione delle cause e degli effetti, dei malfunzionamenti dei servizi in esercizio segnalati dagli utenti interni/esterni. Gli interventi di MAC sono normalmente innescati da una segnalazione di impedimento all'uso dell'applicazione o di una o più delle sue funzioni o dal riscontro di differenze fra l'effettivo funzionamento del software applicativo e quello previsto dalla relativa documentazione o comunque determinato dai controlli che vengono svolti durante l'attività dell'utente, sia esso interno all'Autorità che esterno alla stessa (Stazioni Appaltanti, Cittadini, altre Amministrazioni, ecc.).

L'attivazione dell'intervento avviene attraverso l'apertura di un ticket nell'apposito sistema di tracking (OTRS) da parte dei soggetti autorizzati (funzionari dell'Autorità, personale del support center, ecc.).

Il fornitore traccia l'attività svolta per la risoluzione dell'anomalia sul sistema di bug tracking (redmine).

Gli interventi MAC adottano il ciclo di sviluppo unico nel quale, normalmente, la fase di definizione non viene utilizzata in quanto:

- Gli interventi MAC non prevedono la variazione dei Requisiti Utente
- Gli interventi di MAC sono di norma svolti in garanzia o attraverso strumenti di supporto continuativo da parte del fornitore.

4.5.3 Manutenzione adeguativa

Il servizio di Manutenzione Adeguativa (MAD) si riferisce all'attività volta ad assicurare il costante adeguamento delle procedure e dei programmi all'evoluzione dell'ambiente tecnologico del sistema informativo ed al cambiamento dei requisiti infrastrutturali.

L'attivazione degli interventi MAD viene decisa dall'Autorità sulla base del piano di aggiornamento dei sistemi. Tali interventi seguono uno dei cicli previsti dalle presenti linee guida, stabilito in funzione della complessità dell'intervento. La fase di definizione è avviata senza il coinvolgimento di un ufficio committente, non essendo necessaria la raccolta di nuovi Requisiti Utente. Il piano di lavoro dell'intervento viene redatto dal fornitore ed approvato dagli uffici UPSI e UESI.

5 Linee guida per lo sviluppo delle applicazioni

5.1 Requisiti di carattere generale

Si indicano i seguenti requisiti di massima, di carattere generale, che devono essere rispettati nello sviluppo delle applicazioni e dei servizi:

Codice requisito	Descrizione
REQ.CG.01	Ogni componente realizzato (classe, metodo, variabile, vista, function, ecc) dovrà essere commentata in modo da renderne facilmente individuabile il compito svolto.
REQ.CG.02	Il nome di ogni componente realizzato (classe, metodo, variabile, tabella, vista, function, ecc) dovrà utilizzare la notazione italiana.
REQ.CG.03	Ogni applicazione o servizio deve gestire una KMS degli errori assegnandogli un codice univoco ed una descrizione che identifica la possibile causa e l'eventuale soluzione. IL messaggio restituito dovrà contenere il codice transazione ed il codice identificativo dell'errore.
REQ.CG.04	Al momento della consegna del software, quanto realizzato deve essere compatibile con la release attiva negli ambienti in cui è previsto il deployment.

Tabella 7 - Requisiti di carattere generale

5.1.1 Requisiti di versionamento

Ogni applicazione, servizio o componente realizzata/o deve essere identificata/o da un numero di versione secondo le seguenti regole:

Codice requisito	Descrizione
REQ.CGVERS.01	Le regole per il versionamento suddividono il numero di versione in major number, minor number e revision number con le seguenti regole per l'identificazione della versione: • Il "major number" deve essere variato se vengono aggiunte nuove funzionalità core; • Il "minor number" deve essere variato se vengono aggiunte funzionalità minori e correzioni significative; • Il "revision number" deve essere variato se vengono corretti bug.

REQ.CGVERS.02	Deve essere sempre presente nelle pagine web delle applicazioni il numero di versione attualmente esercita
REQ.CGVERS.03	Ogni componente applicativa facente parte di un servizio deve stampare al suo avvio sui log il proprio numero di versione

Tabella 8 - Requisiti generali per le regole di versionamento

5.1.2 Requisiti per il logging

.

Codice requisito	Descrizione
REQ.CGLOG.01	Tutte le applicazioni ed i servizi realizzati devono loggare in maniera esaustiva le operazioni effettuate in modo che sia possibile ricostruire le attività dell'utente
REQ.CGLOG.02	Ogni scrittura applicativa nel log deve avere il seguente formato: <codicetransazione>-<codicesessione>-{messaggio di testo} dove: • Il codice transazione indentifica la singola request dell'utente • Il codice sessione identifica la sessione applicativa dell'utente Se il codici transazione o il codice sessione non sono presenti devono essere generati dall'applicazioni
REQ.CGLOG.03	Per ciascun errore/eccezione tracciare a livello error il messaggio di errore e gli oggetti rilevanti che hanno causato l'anomalia
REQ.CGLOG.04	Devono essere tracciati a livello di debug le query, i parametri di invocazione di un servizio, i parametri di apertura di una connessione
REQ.CGLOG.05	Nei log non devono essere presenti informazioni sensibili quali nominativo dell'utente, dati carta di credito ecc.

Tabella 9 - Requisiti generali per il logging

5.1.3 Requisiti per l'audit

Codice requisito	Descrizione

REQ.CGAUDIT.01	Ogni operazione di inserimento, modifica e cancellazione deve essere tracciata secondo le specifiche definite dall' Autorità

Tabella 10 - Requisiti generali per il audit

5.1.4 Requisiti per il testing

Per ogni applicazione o servizio devono essere rispettati i seguenti requisiti per il testing:

Codice requisito	Descrizione
REQ.CGTEST.01	Per ogni caso d'uso, relativo ad applicazioni o servizi, deve essere realizzato un test automatizzato che consenta di verificare la sua corretta implementazione.
REQ.CGTEST.02	Per ogni modulo facente parte di una applicazione o servizio devono essere realizzati dei test di unità che consentano di verificare la sua corretta implementazione.
REQ.CGTEST.03	Devono essere realizzati servizi mock, ove non disponibili, per simulare il comportamento di un altro oggetto che durante il test sarebbe difficile o oneroso da istanziare

Tabella 11 - Requisiti generali per il testing

5.1.5 Requisiti di sicurezza

Codice requisito	Descrizione
REQ.CGSEC.01	Le applicazioni ed i servizi realizzati dovranno rispettare i requisiti di sicurezza dettati dal framework OWASP (Open Web Application Security Project)

Tabella 12 - Requisiti di sicurezza

5.2 Requisiti per lo sviluppo Java

Codice requisito	Descrizione
REQ.JAVA.01	Il codice java realizzato per le applicazioni ed i servizi dovrà rispettare gli stili definiti nel documento "Google Java Style" (RIF.EXT.011) ove non diversamente specificato dal presente documento.

REQ.JAVA.02	I nomi dei **package** devono avere la seguente struttura: it.anticorruzione.<nome_progetto>.<nome_sottoprogetto>.<funzionalità>.<sotto_funzionalità>….. dove <nome_progetto> e <nome_sottoprogetto> devono essere nella forma minuscola e <funzionalità> e <sotto_funzionalità> devono essere composte solo da lettere minuscole e la separazione tra parole deve essere ottenuta attraverso la lettera maiuscola della parola che segue (notazione camelCase). Esempio: it.anticorruzione.simog.appalti.gestioneDati.gestioneDatiLotto
REQ.JAVA.03	Ogni applicazione o servizio dovrà essere realizzato come progetto Maven al fine di descriverne in modo dichiarativo come è composto il progetto, quali sono le sue dipendenze, come deve essere compilato e distribuito.
REQ.JAVA.04	Devono essere implementati per tutte le classi che contengono la logica di business dell'applicazione o del servizio i relativi test unitari utilizzando il framework junit. Per testare il corretto funzionamento dei CDI bean si richiede l'utilizzo del framework Arquilian.
REQ.JAVA.05	Utilizzare per il logging la libreria log4j e non la System.out
REQ.JAVA.06	Sostituire sempre toString () con la concatenazione degli attributi per dare una buona rappresentazione dell'oggetto.
REQ.JAVA.07	Il codice deve contenere i commenti scritti secondo le regole definte in **RIF.EXT.011**, utili alla produzione automatica di documentazione javadoc

Tabella 13 - Requisiti Java

5.3 Linee guida sviluppo 'Presentation layer'

Si indicano i seguenti requisiti di massima che devono essere rispettati nello sviluppo dei servizi per i siti e portali Internet e Intranet dell'Autorità:

5.3.1 Requisiti generali

Codice requisito	Descrizione
REQ.GPL.01	Deve essere garantita, laddove possibile, l'aderenza alle raccomandazioni del World Wide Web Consortium (W3C) [HTTP 1.1, HTML 5.0, XHTML (eXtended Hypertext Markup Language) 5, e CSS 3.0]

REQ.GPL.02	Deve essere garantita, laddove possibile, la compatibilità con i browser: Internet Explorer 7.x o superiori, Safari 5.x o superiori, Firefox 20.0 o superiori (obbligatori); Opera 12.0 o superiori (raccomandato); Google Chrome 30.0 o superiori (raccomandato)

Tabella 14 - Requisiti di carattere generale presentation layer

5.3.2 Requisiti di Accessibilità

Codice requisito	Descrizione
REQ.APL.01	Deve essere garantito, laddove possibile, il pieno rispetto della normativa italiana in materia di accessibilità come definito nei RIF.NORM_01,02,03,04,05
REQ.APL.02	Deve essere garantito, laddove possibile, il pieno rispetto delle raccomandazioni W3C (WCAG 2.0- Web Content Accessibility Guidelines) e standard internazionali ISO in materia di accessibilità

Tabella 15 - Requisiti accessibilità presentation layer

5.3.3 Requisiti di Usabilità

Codice requisito	Descrizione
REQ.UPL.01	Laddove possibile, devono essere realizzate funzioni di help contestuale on-line sulle diverse funzionalità offerte dall'interfaccia. Le funzioni di help potranno fare riferimento ad un "Manuale utente" nei casi previsti o richiesti
REQ.UPL.02	I campi presenti nelle varie interfacce hanno tutti un'etichetta verbale, chiara e non ambigua, che spiega il dato che l'utente deve inserire o selezionare. In alcuni casi è possibile che venga richiesto di inserire ulteriori testi al fine di spiegare meglio il dato richiesto come punto fisso oppure a comparsa attivabile dall'utente
REQ.UPL.03	Per ogni form deve essere evidenziato quali informazioni (campi) sono obbligatorie
REQ.UPL.04	Per quanto non già specificato dai requisiti precedenti devono essere garantiti, laddove possibile, il pieno rispetto degli standard internazionali ISO per l'usabilità

Tabella 16 - Requisiti usabilità presentation layer

5.4 Requisiti di testing

Codice requisito	Descrizione

REQ.TPL.01	I test dovranno essere realizzati attraverso il prodotto selenium HQ al fine di consentirne l'esecuzione automatizzata come test JUnit

Tabella 17 - requisiti testing presentation layer

5.5 Requisiti per lo sviluppo OpenData

Codice requisito	Descrizione
REQ.OD.01	I servizi sviluppati in open data dovranno rispettare le linee guida nazionali per la valorizzazione del patrimonio informativo emanate dall'AGID RIF.LGS_03
REQ.OD.02	I formati aperti da mettere a disposizione per la pubblicazione dei dati pubblici al fine di favorirne l'interoperabilità sono json, csv, xml
REQ.OD.03	Ogni dataset pubblicato deve essere accompagnato dai metadati che li descrivano
REQ.OD.04	I dati aperti devono essere disponibili secondo i termini di una licenza o specifiche note di licenza

Tabella 18 - Requisiti sviluppo open data

5.6 Linee guida sviluppo 'Cooperation layer'

Si indicano i seguenti requisiti di massima che devono essere rispettati nello sviluppo dei servizi presenti nel cooperation layer:

5.6.1 Requisiti generali

Codice requisito	Descrizione
REQ.GCL.01	Per quanto riguarda la cooperazione applicativa con altri enti da parte dell'Autorità, nello sviluppo dei servizi si dovrà garantire la conformità con le regole tecniche definite in RIF.LGS_02
REQ. GCL.02	I servizi di cooperazione applicativa, ove possibile devo prevedere nel payload di input un codice identificativo della transazione che verrà utilizzato dal layer sottostante come da requisito REQ. GBL.03

Tabella 19 - Requisiti sviluppo cooperation layer

5.7 Linee guida sviluppo 'Business Process layer'

5.7.1 Requisiti generali

Codice requisito	Descrizione
REQ.GBPM.01	I processi devono essere modellati secondo lo standard BPMN 2.0

Tabella 20 - Requisiti di carattere generale per il business process layer

5.8 Linee guida sviluppo 'Business Rules layer'

5.8.1 Requisiti generali

Codice requisito	Descrizione
REQ.GBRMS.01	Le regole devono essere scritte nel linguaggio Drools

Tabella 21 - Requisiti di carattere generale per il business rules layer

5.9 Linee guida sviluppo 'Integration layer'

5.9.1 Requisiti generali

Codice requisito	Descrizione
REQ.GIL.01	L'integrazione tra i servizi deve essere progettata e implementata attraverso i pattern di integrazione Camel.
REQ.GIL.02	I servizi OSGI devono essere implementati utilizzando il framework blueprint

Tabella 22 - Requisiti di carattere generale per l'integration layer

5.10 Linee guida sviluppo 'Business service layer'

5.10.1 Requisiti generali

Si indicano i seguenti requisiti di massima che devono essere rispettati nello sviluppo dei servizi del business service layer:

5.10.2 Requisiti generali

Codice requisito	Descrizione
REQ.GBL.01	Un servizio esiste solo se implementa almeno un caso d'uso individuato
REQ. GBL.02	Per ogni operation di scrittura deve esistere il corrispettivo servizio di validazione che ha il compito di verificare la correttezza dei dati ricevuti. Questo servizio può essere richiamato direttamente per verificare il set dei dati ricevuti in input e viene chiamato dal servizio stesso prima di procedere con le operazioni di business
REQ. GBL.03	Ad ogni operation del servizio deve essere passato in input un oggetto che contiene: • Il codice transazione relativo all'operazione utente necessario per le attività di logging (richiesto) • L'ID di sessione utente necessario per le attività di logging (opzionale) • La busta SAML necessaria per le attività di audit (opzionale)
REQ. GBL.04	I servizi devono gestire la paginazione dei dati restituiti
REQ. GBL.05	I servizi devono implementare tecniche di caching per l'accesso ai dati più frequenti e con minor frequenza di aggiornamento.

Tabella 23 - Requisiti di carattere generale per il business layer

5.10.3 Requisiti di versionamento

Attraverso l'adozione dei design pattern Canonical Versioning di tipo Strict (**RIF.EXT.010)** si adotta un'unica convenzione di versionamento per i business service, dove ogni cambiamento compatibile o

incompatibile implica una nuova versione del contratto di servizio con la mancanza di retro compatibilità o compatibilità in avanti.

Codice requisito	Descrizione
REQ.VBL.01	Le regole per il versionamento suddividono il numero di versione in major number, minor number e revision number con le seguenti regole per l'identificazione della versione: • I cambi della signature delle operation, l'aggiunta di nuove operation e la modifica della semantica di operation pre-esistenti comportano l'aggiornamento del numero di versione major; • L'aggiunta di funzionalità minori e correzioni significative comportano l'aggiornamento del minor number; • La correzione di bug comporta l'aggiornamento del revision number.

Tabella 24 - Requisiti per le regole di versionamento

Inoltre sono definite delle politiche di dismissione dei servizi non più utilizzati per evitare il proliferare degli stessi.

5.10.4 Requisiti di nomenclatura

Le convenzioni di nomenclatura sono un insieme di regole con cui identificare, attraverso nomi logici, tutti i servizi ANAC. Tutti i servizi ANAC devono avere un nome quanto più possibile auto-esplicativo della funzionalità di business che offre all'interno del dominio ANAC.

Di seguito i requisiti di massima, relativi alle convenzioni di nomenclatura, che devono essere soddisfatti nel momento della creazione di un servizio:

Codice requisito	Descrizione
REQ.NBL.01	Nel nome di un servizio/operation devono essere omessi proposizioni, articoli, caratteri speciali
REQ.NOM.02	Nel nome di un servizio/operation non devono essere usati numeri come elementi di distinzione o per distinguere le diverse versioni ad eccezione di quanto richiesto dal pattern di versionamento.

REQ. NBL.02	Il nome di un **servizio:** a. deve iniziare con la lettera **maiuscola;** b. deve essere composto solo da lettere minuscole e la separazione tra parole deve essere ottenuta attraverso la lettera maiuscola della parola che segue (notazione CamelCase); c. non sono ammessi blank o, di norma, caratteri underscore. d. Il nome del servizio deve essere significativo per la funzionalità che eroga e. Il nome di un servizio deve essere composto dalla coppia <sostantivo><nome> dove: 1. <sostantivo> indica lo scopo del servizio ad esempio Gestione 2. <entità> rappresenta una delle entità di business presenti nel modello dei dati del dominio ANAC. (es: Gara, Lotto, StazioneAppaltante)
REQ. NBL.03	Il nome delle **operation** del servizio: a. deve iniziare con la lettera **minuscola** b. deve essere composto solo da lettere minuscole e la separazione tra parole deve essere ottenuta attraverso la lettera maiuscola della parola che segue (notazione camelCase); c. non sono ammessi blank o, di norma, caratteri underscore. d. Il nome dell'operation deve essere significativo per la funzionalità che eroga e. deve essere composto dalla coppia <verbo><nome> dove: a. <azione> 1. può essere un verbo CRUD (Create, Read, Update, Delete) nella notazione italiana (Crea, Consulta, Modifica, Cancella etc etc) 2. un verbo specifico del dominio ANAC (es: Perfeziona) 3. un verbo indicativo di una verifica (es: Verifica) b. <entità> un sostantivo che rappresenta una delle entità di business presenti nel modello dei dati del dominio ANAC. (es: Gara, Lotto, StazioneAppaltante)
REQ. NBL.04	Le **operation** possono essere distinte in due insiemi: a. Operation che lavorano su di un singolo elemento (sia lettura che scrittura) b. Operation che lavorano su di una lista di elementi (sia lettura che scrittura)

	e nel qual caso al nome viene aggiunta la stringa '**Lista'** Esempi di nomi di operation: consulta**G**ara (tipo a), inserisci**L**ista**S**tazioni**A**ppaltanti (tipo b)

Tabella 25 - Requisiti per le convenzioni di nomenclatura

5.10.5 Servizi di accesso alla BDNCP

I '*Servizi BDNCP*' sono un sottoinsieme dei '*Servizi di business* ed hanno la responsabilità di essere l'unico punto di accesso, alla banca dati BDNCP.

Partendo dalla ripartizione dei dati in macro-aree fatta in BDNCP esiste una tassonomia dei '*Servizi BDNCP*' suddivisa in quattro aree funzionali, ognuna con caratteristiche diverse a seconda dei dati trattari. Di seguito una figura che mostra le quattro tassonomie dei servizi:

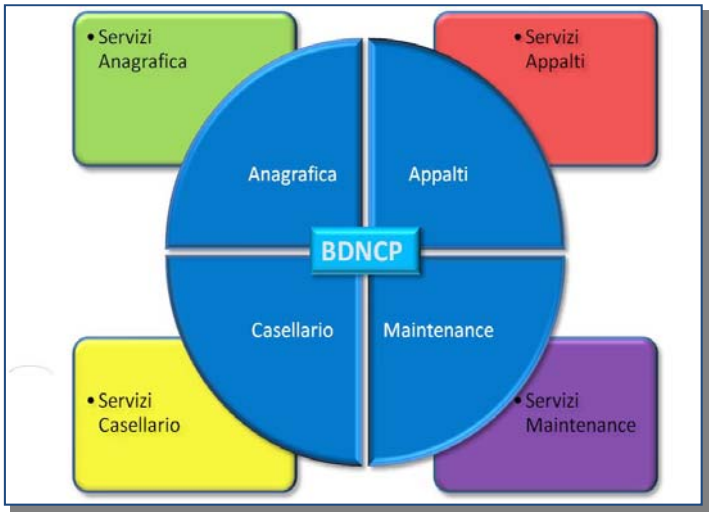

Figura 11 - Tassonomia dei *'Servizi BDNCP'*

dove:

- ***Servizi Anagrafica:*** contiene i servizi per l'accesso alle componenti dell'anagrafica. In tale insieme possono essere inclusi i servizi di creazione e gestione delle utenze associate alle stazioni appaltanti;
- ***Servizi Appalti:*** contiene i servizi per la gestione del ciclo di vita di un appalto. In tale insieme possono essere inclusi i servizi di creazione e gestione della gara e dei lotti;
- ***Servizi Casellario:*** contiene i servizi per la gestione delle funzionalità del Casellario. In tale insieme possono essere inclusi i servizi di creazione e gestione dei CEL;

- ***Servizi Maintenance:*** contiene i servizi per la gestione dei dati presenti in BDNCP. In tale insieme possono essere inclusi i servizi di creazione e gestione delle tipologiche.

Successivamente alla classificazione dei servizi, sono stati introdotti ulteriori vincoli logici sulla definizione degli stessi, in particolare nella realizzazione di un servizio devono essere rispettati i seguenti requisiti:

Codice requisito	Descrizione
REQ.SBDNCP.01	Un servizio può accedere in scrittura solo alla sua area funzionale di BDNCP (es: anagrafica)
REQ.SBDNCP.02	Un servizio può accedere in lettura solo alla sua area funzionale di BDNCP (questa regola generale può avere delle eccezioni)
REQ.SBDNCP.03	Un servizio può accedere ai dati riferiti ad un'altra area funzionale di BDNCP solo attraverso servizi di pertinenza di quell'area.

Tabella 26 - Requisiti per i servizi BDNCP

5.11 Linee guida sviluppo 'Data Layer'

Si indicano i seguenti requisiti di massima che devono essere rispettati nello sviluppo dei servizi del data layer.

5.11.1 Requisiti generali

Codice requisito	Descrizione
REQ.GDB.01	Sono stati individuati quattro connection pool che i servizi dovranno utilizzare per l'accesso alla BDNCP. In particolare ogni servizio deve utilizzare il pool che accede all'area funzionale di competenza: 1. Pool di accesso all'area **Appalti** 2. Pool di accesso all'area **Anagrafica** 3. Pool di accesso all'area **Casellario** 4. Pool di accesso all'area **Maintenance**

Tabella 27 - Requisiti di carattere generale per il data layer

5.11.2 Requisiti di nomenclatura

Codice requisito	Descrizione
REQ.DBCN.01	I nomi di schema, tabelle, colonne, stored_procedure, function, trigger, viste e altri oggetti devono essere descrittivi (es. non X25TZ ma es. PRO.NULLA_OSTA)
REQ.DBCN.02	I nomi per schema, tabelle, colonne e viste devono essere maiuscoli
REQ.DBCN.03	I nomi senza spazi ma con "_" tra parole
REQ.DBCN.04	I nomi delle stored procedure iniziano per "sp_"
REQ.DBCN.05	I nomi delle viste iniziano per "v_"
REQ.DBCN.06	I nomi delle funzioni iniziano per "f_"
REQ.DBCN.07	Non riutilizzare nomi già in uso in altri schema

Tabella 28 - Requisiti per le convenzioni di nomenclatura del database

5.11.3 Requisiti di modellazione dei dati

Codice requisito	Descrizione
REQ.DDBV.01	Tutte le tabelle devono avere chiave primaria
REQ.MDDBV.02	Le chiavi di tabella di tipo bigint (con identity), salvo utilizzi specifici che si dovranno giustificare
REQ.MDDBV.03	Le tabelle dovranno essere in terza forma normale, salvo per le applicazioni di business intelligence o per specifici utilizzi che si dovranno giustificare
REQ.MDDBV.04	Se necessario, creare per lo specifico contesto uno schema nel quale allocare gli oggetti dello specifico dominio

REQ.MDDBV.05	Non duplicare colonne e tabelle esistenti
REQ.MDDBV.06	Non utilizzare nomi corrispondenti a parole riservate TSQL 2005/2008
REQ.MDDBV.07	Le colonne che contengono date dovranno essere del tipo datetime
REQ.MDDBV.08	Non memorizzare documenti all'interno del database ma utilizzare collegamenti a filesystem (2005) o tecnologia filestream (2008), salvo specifici utilizzi che si dovranno giustificare
REQ.MDDBV.09	Le colonne che hanno un valore di default devono avere il vincolo NOT NULL
REQ.MDDBV.10	Le colonne collegate con foreign key non devono avere valori NULL, salvo documentate esigenze
REQ.MDDBV.11	Evitare l'uso del tipo text, utilizzare il tipo VARCHAR(MAX)
REQ.MDDBV.12	Evitare l'uso dei tipi di dati UNICODE, NVARCHAR e NCHAR. Si valuteranno specificamente eccezioni documentate
REQ.MDDBV.13	Necessario progettare adeguatamente le strutture di indicizzazione in funzione delle consuete modalità di utilizzo dei dati (non è un requisito, ma è opportuno che si ponga attenzione di queste strutture in funzione degli utilizzi dei dati stessi)
REQ.MDDBV.14	Le colonne di tipo stringa/alfanumerici dovranno utilizzare il tipo varchar o nvarchar, salvo specifici utilizzi che si dovranno giustificare
REQ.MDDBV.15	La lunghezza dei campi deve essere sufficiente al contenimento dei dati ma non sovrabbondante (es. per il CAP il tipo corretto è varchar(5) non varchar(50))
REQ.MDDBV.16	Non ricreare oggetti che semanticamente sono superflui perché analoghi sono già presenti in altri schema

REQ.MDDBV.17	Tutte le stored procedure, funzioni e trigger devono avere come primo statement SET NOCOUNT ON per garantire prestazioni ottimali
REQ.MDDBV.18	Nelle stored procedure non utilizzare l'opzione RECOMPILE
REQ.MDDBV.19	Mettere sempre le istruzioni di DECLARE sempre all'inizio del codice di procedure e funzioni
REQ.MDDBV.20	I parametri delle stored procedure e function devono avere esattamente lo stesso tipo delle colonne di riferimento
REQ.MDDBV.21	Nelle condizioni di SELECT dirette o tramite VISTE i parametri utilizzati devono essere esattamente dello stesso tipo delle colonne di riferimento
REQ.MDDBV.22	Nelle istruzioni di SELECT non occorre mai utilizzare il carattere jolly *
REQ.MDDBV.23	Cercare di evitare i cursori lato server
REQ.MDDBV.24	Cercare di evitare l'utilizzo di tabelle temporanee
REQ.MDDBV.25	Cercare di evitare i caratteri jolly nelle like in testa o in mezzo a token di ricerca (es. where nomecampo like '%test', where nomecampo like 'test%ami')
REQ.MDDBV.26	Cercare di evitare gli operatori di negazione nelle condizioni di ricerca NOT e <> poichè determinano quasi sempre table scan
REQ.MDDBV.27	Cercare di non superare in unico oggetto un numero di join superiore a 4. Casi particolari andranno vallutati
REQ.MDDBV.28	Le operazioni di ordinamento dei risultati dovranno essere a carico dei componenti applicativi. Eventuali eccezioni dovranno essere documentate e approvate
REQ.MDDBV.29	Le operazioni di raggruppamento dei dati dovranno essere a carico dei componenti applicativi. Eventuali eccezioni dovranno essere documentate e approvate

REQ.MDDBV.30	Iniziare il codice di stored procedure, batch e trigger con il comando SET NOCOUNT ON.
REQ.MDDBV.31	Utilizzare IF EXISTS piuttosto che SELECT COUNT(*) per verificare presenza di record
REQ.MDDBV.32	Se nel codice è necessario utilizzare funzioni di sistema come suser_name() o getdate() più volte, memorizzare il risultato in una variabile e riutilizzare lo stesso invece di richiamare le stesse alla esecuzione multipla
REQ.MDDBV.33	Utilizzare il corretto livello di isolamento in funzione della tipologia di accesso necessario
REQ.MDDBV.34	Evitare transazioni complesse e lunghe nei trigger
REQ.MDDBV.35	Analizzare i piani di esecuzione ed modellare gli oggetti in modo tale da eliminare attività di table o index scan su tabelle con molti dati
REQ.MDDBV.36	Non utilizzare comandi DDL in stored procedure/function
REQ.MDDBV.37	Definire obbligatoriamente il parametro di LOCK_TIMEOUT per le connessioni a 3 minuti. Eventuali eccezioni andranno opportunamente giustificate ed autorizzate
REQ.MDDBV.38	Definire obbligatoriamente il parametro di QUERY TIMEOUT per le connessioni a 3 minuti. Eventuali eccezioni andranno opportunamente giustificate ed autorizzate

Tabella 29 - Vincoli sul database

6 Versione dei prodotti utilizzati

Codice	Tipologia	Prodotto	Versione
RIF.PROD.001	Sistema operativo	Red Hat Enterprise Linux	6

RIF.PROD.002	JVM	Oracle JDK	1.6
RIF.PROD.003	Application server	JBoss EAP	5.1.2
RIF.PROD.004	Portal server	JBoss EPP	5.2.2
RIF.PROD.005	ESB	JBoss Fuse	6.2
RIF.PROD.006	BPM	JBoss BPM	6.1
RIF.PROD.007	BRMS	JBoss BRMS	6.1
RIF.PROD.008	ECM	EXO Portal	3.5
RIF.PROD.009	Javascript MVW framework	Angular JS	1.2.x
RIF.PROD.010	ECM	Microsoft Sharepoint	2010
RIF.PROD.011	DBMS	Microsoft SQL Server Enterpise edition	2008 R2
RIF.PROD.012	ECM	Alfresco Enterprise edition	4.1.4
RIF.PROD.013	BI	Microstrategy Reporting Suite	9

Tabella 30 - Prodotti e tecnologie utilizzate