



Agile methodology design guidelines

Linee guida progettazione metodologia Agile

Version: 4.0
Date: 16/10/2017

ABSTRACT

Description of the methods characterizing the Agile development methodology within the context of Cloudify NoiPA. The document is an update of the guidelines previously drafted, "PMO_Linee Design Guide Agile_v2.0 Methodology", defined on the basis of the outputs, shared and approved, during Cloudify NoiPA workshops. The current version of the document aims at acquiring the changes occurred during the course of the first release in terms of roles and responsibilities (introduction of the role of the PPO - Proxy Product Owner) with particular reference to requirements management of the sprint review phase (See PMO_20170912_Sprint Review_V2.0.ppx). The section of the document dedicated to the Foundation development phase has not been updated since it has already been implemented.

The Title and Abstract of the document have been translated with the aim to summarize the themes analyzed.

The following content will be presented instead in its original version (Italian)

1. DESCRIZIONE DEL PROGETTO

1.1 ABSTRACT

Descrizione delle modalità per le quali si caratterizza la metodologia di sviluppo Agile all'interno del contesto di Cloudify NoiPA..

In particolare il documento rappresenta un aggiornamento delle linee guida precedentemente redatte, "PMO_Linee Guida Progettazione Metodologia Agile_v2.0", documento definito sulla base degli output, condivisi e approvati, nel corso dei workshop svolti sul tema Cloudify NoiPA. L'attuale versione del documento è finalizzata ad acquisire le variazioni intervenute nel corso dello svolgimento della prima release in termini di ruoli e responsabilità (introduzione del ruolo del PPO – Proxy Product Owner), con particolare riferimento alla gestione dei requisiti e alla fase di sprint review (Vedi *PMO_20170912_Sprint Review_V2.0.pptx*).

La sezione del documento dedicata alla fase di sviluppo Foundation non ha subito aggiornamenti essendo la stessa già stata realizzata.

1.2 OBIETTIVI DEL DOCUMENTO

L'obiettivo è quello di definire e condividere le linee guida organizzative e di processo per lo svolgimento delle attività propedeutiche all'attivazione degli sviluppi e per la realizzazione degli sviluppi stessi.

Le tematiche oggetto di tale documento sono:

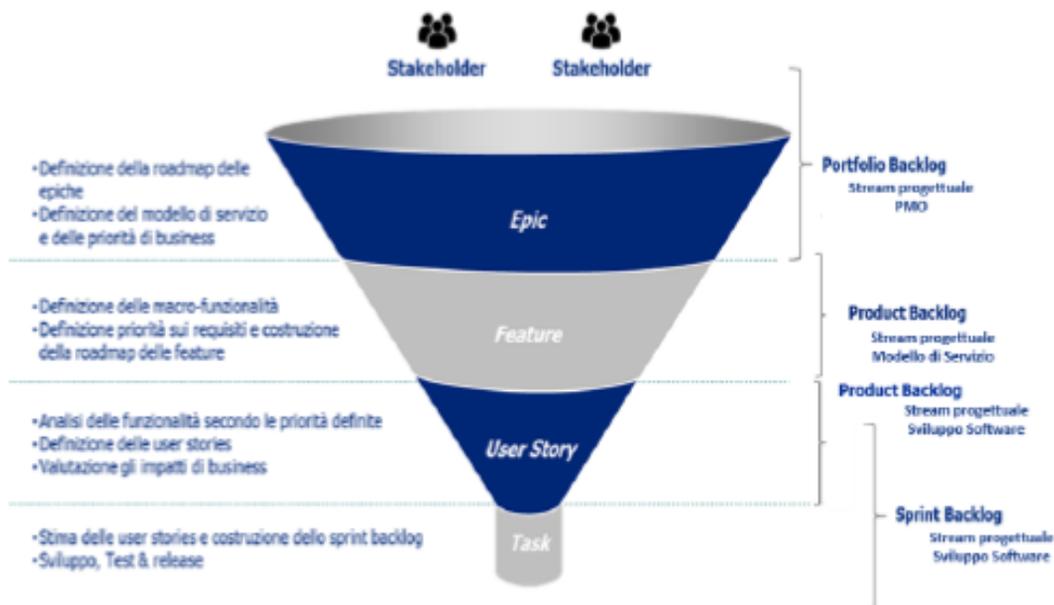
- La metodologia Agile per Cloudify NoiPA
- Il processo dalla pianificazione del requisito all'erogazione del servizio
- Il modello organizzativo in Team
- Il processo di gestione dei requisiti
- Il processo dallo sviluppo all'esercizio: ambienti, ruoli e responsabilità

2. LA METODOLOGIA AGILE PER CLOUDIFY NOIPA

La metodologia Agile si basa sui concetti di:

- epic
- feature
- user story
- task

di seguito meglio descritti.



Il ciclo standard di sviluppo Agile si costituisce delle macro-fasi:

- Foundation
- Portfolio Management
- Release Planning (Epica)
- Release review
- Release retrospective

La fase di **Foundation**, è costituita da tutte le attività preparatorie all'avvio dello sviluppo, in questa fase i principali task svolti sono:

- identificare le epiche
- definire il modello di servizio e la UX
- definire una mappa applicativa del prodotto
- definire tutte le linee guida sw
- definire la roadmap delle epiche

La fase di **Portfolio Management**, è costituita da tutte le attività di pianificazione della release, in questa fase i principali task svolti sono:

Report

- pianificare le release (a livello di epiche);
- verificare le tempistiche di implementazione delle release;
- aggiornare la roadmap delle release da sviluppare.

La fase di **Release Planning**, è costituita da tutte le attività di definizione e preparazione degli sprint, in questa fase i principali task svolti sono:

- aggiornare gli output della fase di foundation;
- rivedere la Definition of done;
- definire il modello logico dei dati;
- dettagliare le feature della release;
- dettagliare i requisiti attraverso le user story;
- definire i criteri di accettazione utente;
- richiedere e valutare le stime economiche;
- pianificare le user story negli sprint (sprint backlog);
- assegnare i ruoli (product owner, scrum master,...) a livello di sprint.

Durante la Release si eseguono uno o più **sprint**, le fasi di ogni sprint sono:

- **Planning**: si dettagliano i task tecnici e si definiscono delle priorità
- **Execution**: si produce il software per le user story relative e si eseguono i test
- **Review**: si presenta il prodotto dello sprint e si testa la qualità del sw
- **Retrospective**: si individuano delle azioni di miglioramento per il dev team

Durante la fase di **Release Review** si esaminano (ispezione) i prodotti dei diversi sprint della release, in particolare le attività effettuate sono:

- collaudare il sw prodotto nella release (cross sprint);
- presentare quanto realizzato;
- testare la qualità sw;
- rilasciare in ambiente collaudo;
- rilasciare in ambiente pre-esercizio (o esercizio).

Durante la fase di **Release Retrospective** si analizzano i fattori causali e si identificano eventuali azioni di miglioramento. Dopo la release retrospective il portfolio management aggiorna il portfolio delle release e la roadmap.

Si riporta di seguito uno schema rappresentativo del modello concettuale di interazione tra le fasi del processo di sviluppo Agile e i diversi stadi di rilevazione, elaborazione e gestione dei requisiti.

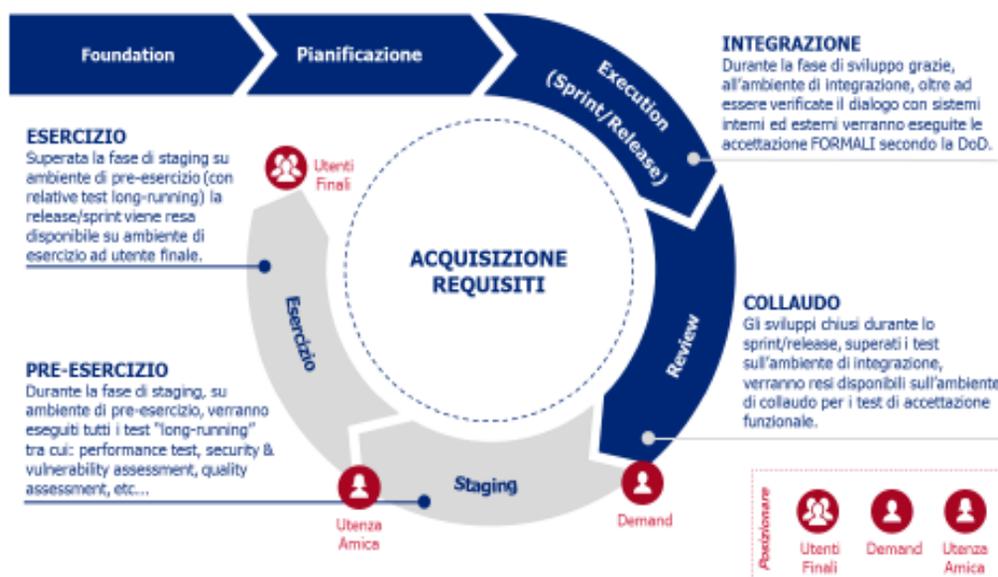


3. IL PROCESSO DALLA PIANIFICAZIONE DEL REQUISITO ALL'EROGAZIONE DEL SERVIZIO

Il processo di sviluppo sw in modalità Agile, come di seguito rappresentato, prevede dopo la realizzazione della fase di Foundation, di natura preparatoria e di un successivo ciclo iterativo che deve essere ripetuto tante volte quante sono le release pianificate.

I prodotti dei singoli sviluppi e di conseguenza di ciascuna release assumono uno stato differente a seconda della fase e dell'ambiente in cui si trovano. Inoltre, gradualmente intervengono le diverse tipologie di utenti al fine di verificare la coerenza degli output con i requisiti e con le esigenze espresse:

- il Demand in fase di collaudo;
- l'Utenza amica (Stakholder esterni che collaborano con la DSII nella realizzazione del nuovo sistema) in fase di staging;
- gli Utenti finali in fase di Esercizio (attivazione dei servizi verso le Amministrazioni).



3.1 IL PROCESSO: LA FASE DI FOUNDATION

La fase di Foundation è la prima del processo di sviluppo Agile, essa consiste nell'esecuzione di tutte le attività propedeutiche all'attivazione degli sviluppi veri e propri e quindi alle fasi successive (Portfolio Management, Release, Sprint). In particolare, nel corso di tale fase vengono definito i processi, le linee guida e le modalità operative afferenti i diversi ambiti progettuali.



3.1.1. LA FASE DI FOUNDATION: RUOLI E RESPONSABILITÀ

Si riporta di seguito un elenco dettagliato delle attività identificate per la fase di Foundation e per ciascuna di essa i ruoli coinvolti nello svolgimento delle stesse e le relative responsabilità.

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastrutture	Data Governance	CTO	Linee Guida Qualità SW	PO	PMO	SM	Dev Team
*Tutti i soggetti dovranno essere informati rispetto alle attività.												
Mappe funzionali AS IS		A/R		C								
Mappe funzionali TO BE		A/R		C								
Linee Guida UX/UI		A/R		C								
Architettura funzionale		A/R		C	C							
Requisiti di dettaglio su prima release fino a livello di User Story		A/R		C		C						
Definizione del modello di gestione dei requisiti		R		R			A/R					
Attivazione TFS					A/R							
Esigenze di business legate allo storico		A/R		C	C	C						
Reverse Eng. per Mapping su User Story		C		A/R		C						
PoC di Sistema		C		A/R	R	R						
Architettura applicativa				A/R		C						
Abilitazione ambiente di Integrazione					A/R							
Configurazione MDM e set up applicativo				R	R	A/R						
Configurazione BPM e set up applicativo				A/R	R							
Identificazione framework e linee guida di base SW				A/R								
Linee Guida Architettura					A/R							

*Tutti i soggetti dovranno essere informati rispetto alle attività.

	PM	Modello di servizio	Comunicazione	Sviluppo Software	Infrastrutture	Data Governance	CTO	Linee Guida Qualità SW	PO	PMO	SM	Dev Team
Linee Guida SW				A/R								
Linee Guida Security	R			R	R	R	A					
Linee Guida Qualità SW								A/R				
Linee Guida Dati						A/R						
Disegno di sistema del CTO							A/R					
Modello Concettuale (fisico)						A/R						
Data Glossary						A/R						
Definizione del processo dallo sviluppo all'esercizio	A			R	R	C						
Definizione e disponibilità ambienti					A/R							
Definizione accessi				C	A/R							
Logistica team agile	A/R											
Soluzione contrattuale	A/R			C				R				
Portale strumento di lavoro condiviso	C	C	A/R	C	C	C						
Piano prototipi e mockup		A/R	C	C								
Modello di intermediazione		A/R		C	C	C						
Identificazione stream paralleli	A	C	C	R	R	R	C	C				

3.2 IL PROCESSO DALLA PIANIFICAZIONE ALLA REVIEW

3.2.1 IN/OUT VS FAST

La fase di Foundation è seguita dal ciclo di release. Per ciascuna fase di una generica release_n si evidenziano gli input e gli output previsti dalla metodologia.

In generale ogni release riceve in input i requisiti e fornisce in output la validazione funzionale di quanto sviluppato nel corso della release stessa.

Nell'immagine seguente a titolo esemplificativo sono stati rappresentati soli due sprint per una release. Ciascuno sprint della release riceve in input il piano della release ed eventuali BUG o Debiti tecnici derivanti dallo sprint precedente e fornisce in output la validazione funzionale di quanto sviluppato più eventuali BUG o debiti tecnici.

Release Planning:

- **Input:** Requisiti funzionali release_n
- **Output:** Plan release_n

Sprint Planning

- **Input:** Plan release_n; Debito tecnico sprint_(n-1); **BUG** sprint_(n-1)
- **Output:** Plan sprint_n

Sprint Execution

- **Input:** Plan sprint_n
- **Output:** SW Dev. & Tested (Unit, Style, ...)sprint_n; Debito Tecnico sprint_n

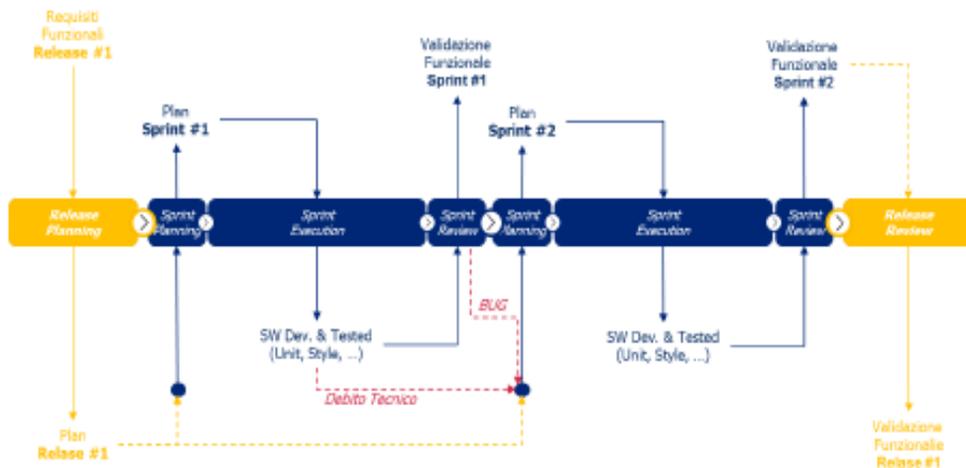
Report

Sprint Review

- *Input:* SW Dev. & Tested (Unit, Style, ...)sprint_n
- *Output:* Validazione Funzionale Sprint n; BUGsprint_n

Release Review

- *Input:* Validazione Funzionale Sprint_n
- *Output:* Validazione Funzionale Release_n

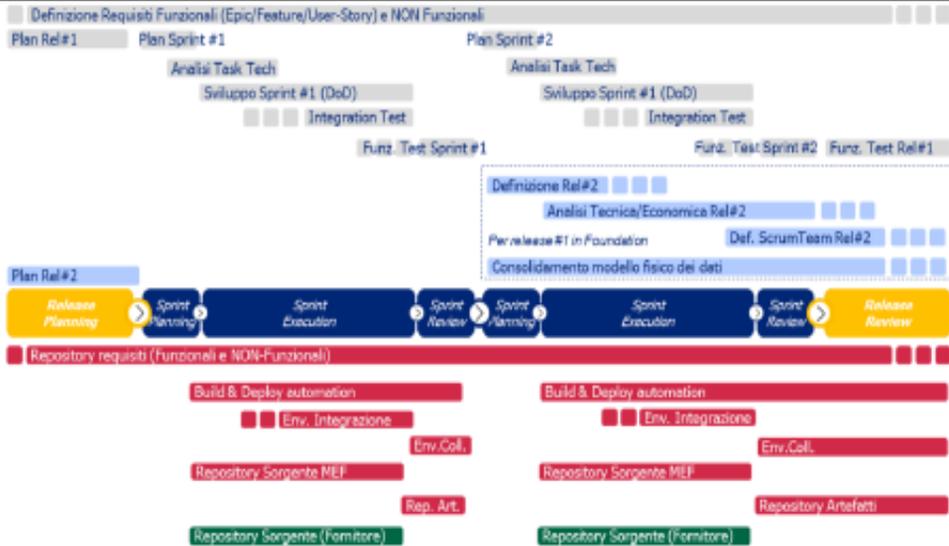


3.2.2 ATTIVITÀ VS STRUMENTI/AMBIENTI

Nell'immagine seguente, si rappresenta una visione d'insieme di tutte le attività che saranno effettuate all'interno di uno sprint e delle attività preparatorie:

- un mese prima dal termine di una generica release_n, devono essere disponibili i requisiti di input previsti per la release successiva, viene quindi attivata la fase preparatoria della release_{n+1}, finalizzata all'esecuzione di tutte le attività di pianificazione e organizzazione necessarie per la fase di Release Planning. Tali attività per la prima release sono effettuate nel corso del Foundation.
- prima dell'inizio della fase di execution della Release viene consolidata la porzione del modello fisico dei dati necessario per lo svolgimento della stessa.
- le attività di test è collaudo vengono effettuate sugli output dei singoli sprint e sull'output integrato dell'intera release.

Report



3.2.3 LA RELEASE: RUOLI E RESPONSABILITÀ

Si riporta di seguito un elenco dettagliato delle attività identificate per le fasi della release. Per ciascuna attività si indicano i ruoli coinvolti nello svolgimento delle stesse, oltre che le relative responsabilità. Le attività evidenziate in azzurro sono quelle di natura preparatoria.

3.2.3.1 Release Planning

*Tutti i soggetti dovranno essere informati rispetto alle attività.

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastrutture	Governance Data	CTO	Linea Guida Qualità SW	PO	PPO	SM	DevTeam
Rivede modello logico Dati Epica		C		C	I	A/R						
Assegna ruoli (PO, PPO, SM, DevTeam) a livello di sprint	A	R		R	C	R						
Rivede gli output della fase di foundation		R		R		R	A/R	R				
Fornisce eventuali dettagli su epica/feature		A/R		R		C						
Fornisce eventuali dettagli su user story		R		A/R								
Fornisce eventuali dettagli su criteri di accettazione su epica/feature		A/R		R								
Fornisce eventuali dettagli su criteri di accettazione su user story		R		A/R								
Definition of done su epica/feature/user story		R		R		R	A/R	R				
Pianifica le user story negli sprint (sprint backlog)		C		A/R		C						

Report

3.2.3.2 Sprint Planning

*Tutti i soggetti dovranno essere informati rispetto alle attività.

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastrutture	Data Governance	CTO	Linee Guida Qualità SW	PO	PRO	SM	Dev/Team
Identifica le priorità di business rispetto ad attività a piano	A/R			C		C			I	I	I	
Pianifica le user story nello sprint (sprint backlog)		C		A/R		I			C	R	R	
Controlla l'effettiva applicazione della DoD (generica) per lo specifico sprint									R	A/R	R	C
Scomponi in task tecnici le user story		I		C		C				R	R	A/R
Verifica l'analisi tecnica delle user story dello sprint (Compresi Task)				C		C				A/R		
Garantisce applicazione linee guida e rispetto standard				C	C	C				A/R		

3.2.3.3 Sprint Execution

*Tutti i soggetti dovranno essere informati rispetto alle attività.

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastrutture	Data Governance	CTO	Linee Guida Qualità SW	PO	PRO	SM	Dev/Team
Scomponi in task tecnici le user story		I		C		C				R	R	A/R
Supporta team di sviluppo nel comprendere a fondo i requisiti		C		C					R	A/R		
Facilita e rimuove impedimenti											A/R	
Garantisce commitment dei dev team											A/R	
Assicura la time box (rispetto tempi e deliverable)											A/R	
Assicura la coesione tra i diversi partecipanti del dev team											A/R	
Assicura corretta integrazione diversi dev team all'interno feature/epica											A/R	
Sviluppa le user story dello sprint (realizzazione codice)				C	C	C					R	A/R
Testing user story: unit, security, integrazione, regression e prestazionale				C	C	C					R	A/R
Garantisce applicazione linee guida e rispetto standard				C	C	C					A/R	
Definizione Epic/Feature su prossima release		A/R		R		R						
Definizione User Story su prossima release		R		A/R		R						
Esegue Analisi Tecnica/Economica su prossima release		C		A							R	R
Valuta organizzazione e composizione scrum team su prossima release		A/R		R								
Consolidamento modello fisico dei dati		C		C		A						R

3.2.3.4 Sprint Review

*Tutti i soggetti dovranno essere informati rispetto alle attività.

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastrutture	Data Governance	CTO	Linee Guida Qualità SW	PO	PRO	SM	Dev/Team
Collauda (funzionalmente) lo sprint (tutte le user story chiuse)									R	A/R	C	C
Presenta criticità sprint per eventuale piano di rientro su debito tecnico									C	R	A/R	C
Supportano e presentano quanto realizzato									R	R	A/R	R
Testing qualità software								C		A/R	R	
Rilascio in ambiente di integrazione/colloquio*												

Per gli approfondimenti di dettaglio della fase di Sprint Review si rimanda al capitolo 7 del presente documento.

3.2.3.5 Sprint retrospective

Analisi fattori causali e identificazione azioni di miglioramento

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastruttura	Data Governance	CTO	Linee Guida Qualità SW	PO	PMO	SM	Dev/Team
Analisi fattori causali e identificazione azioni di miglioramento		R		A/R		I		I	R	R	R	R

3.2.3.6 Release Review

*Tutti i soggetti dovranno essere informati rispetto alle attività.

Collaudo funzionale (cross sprint) fatto su feature e epica

Presenta quanto realizzato

Verifica di qualità (Opzionale)

Testing prestazionale

Rilascio in ambiente di integrazione/collaudo*

Rilascio in ambiente di pre-esercizio/esercizio*

Verifica la pianificazione delle release rispetto alle Epic/Feature chiuse

Aggiorna il piano delle release

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastruttura	Data Governance	CTO	Linee Guida Qualità SW	PO	PMO	SM	Dev/Team
Collaudo funzionale (cross sprint) fatto su feature e epica		A/R										
Presenta quanto realizzato		C		A/R								
Verifica di qualità (Opzionale)				A/R	R							
Testing prestazionale				R	A/R							
Rilascio in ambiente di integrazione/collaudo*												
Rilascio in ambiente di pre-esercizio/esercizio*												
Verifica la pianificazione delle release rispetto alle Epic/Feature chiuse	A/R	R		R		C						
Aggiorna il piano delle release	A/R	C		C		C						

3.2.3.7 Release retrospective

Analisi fattori causali e identificazione azioni di miglioramento

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastruttura	Data Governance	CTO	Linee Guida Qualità SW	PO	PMO	SM	Dev/Team
Analisi fattori causali e identificazione azioni di miglioramento		I		A/R	I	I		I				I

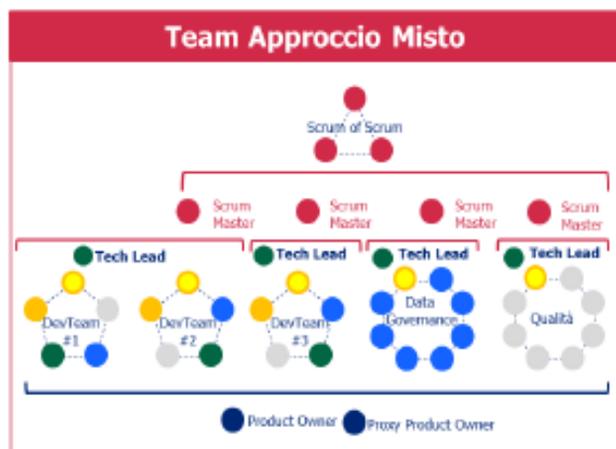
4. MODELLO ORGANIZZATIVO TEAM

Il modello organizzativo dei team identificato per gli sviluppi di Cloudify NoiPA è un modello ibrido, dato dal incrocio tra i modello cross funzionale e specializzato. In particolare, esso è costituito da:

- **team di sviluppo cross funzionali**, caratterizzati da competenze eterogenee (front end, back end, sviluppo sw, accesso ai dati, ecc.), comprese le competenze relative all'ambito dati e alla gestione degli accessi ai dati: questo team sarà responsabile di eseguire lo sviluppo delle user story assegnate dalla parte di front-end sino all'accesso ai dati. Non è di sua competenza la progettazione e realizzazione del modello dati;
- **team specializzati nell'ambito Data Governance** che si occuperanno di progettare e sviluppare il modello dati (logico e fisico), mantenendolo coerente e consistente. Prima dell'avvio dello sprint, la relativa componente dati dovrà essere già realizzata per permettere ai team di sviluppo di lavorare in autonomia;
- **team specializzato nell'ambitoQualità** che definisca le linee guida di qualità del sw
- **technical leader**, un referente tecnico del fornitore verso l'Amministrazione per ciascun dev team o trasversale a più dev team.

L'organizzazione in particolare prevede il seguente posizionamento delle figure di **Scrum Master**, **Product Owner**, **Proxy Product Owner** e **Technical Leader**.

Per ogni linea progettuale, come da RACI, ogni team fornisce indicazioni rispetto all'assegnazione dei ruoli (PO, PPO, SM, Dev Team) a livello di sprint e il PMO provvede alla raccolta delle diverse componenti e al consolidamento delle stesse.



L'approccio selezionato è caratterizzato dai seguenti:

PRO

- Responsabilità chiara su progettazione e realizzazione del modello logico e fisico dei dati;
- Omogeneità e consistenza del modello dati (dal concettuale al fisico) sia in fase di progettazione che di realizzazione;
- Preservata la vista funzionale dell'approccio «cross-funzionale» con responsabilità end to end su intero stack anche attraverso «innesti» di figure esperte su Data Governance sui DevTeam.

CONTRO

- Necessario prevedere una corretta sincronizzazione delle attività tecniche di sviluppo tra i DevTeam e i team di Data Governance;
- Se non correttamente dimensionato il team di Data Governance potrebbe generare inefficienza o ritardo sulle attività (collo di bottiglia sui DevTeam);
- Aumento della complessità nello sviluppo/quotazione di requisiti di business che prevedono la sovrapposizione di elementi tecnici sia sulla componente dati (Data Governance) che di sviluppo (DevTeam).

5. IL PROCESSO DI GESTIONE DEI REQUISITI

Il modello di gestione dei requisiti indentificato si basa:

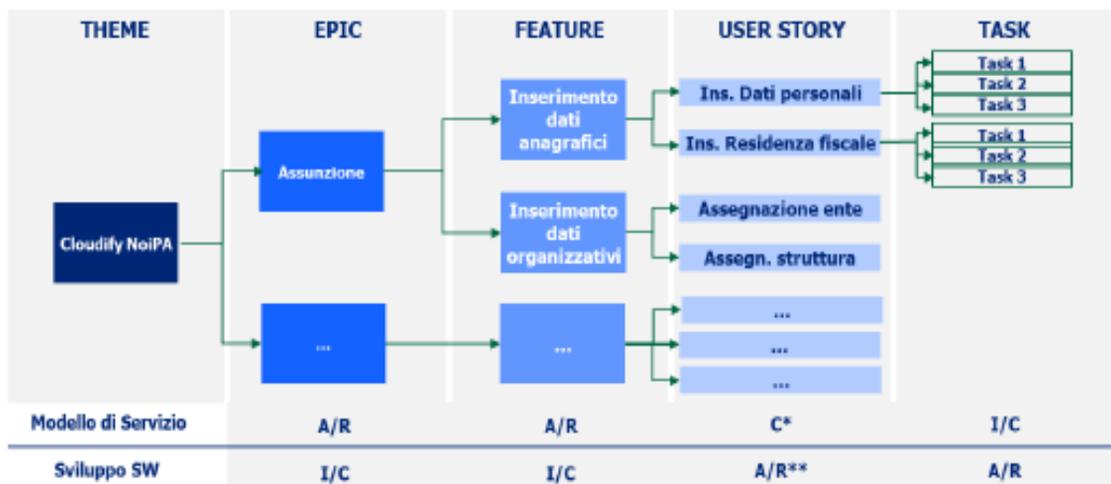
- su una logica per processo;
- sul concetto di requisito «SMART», regolamentata per la stesura e la modellazione (UML);
- su un **approccio gerarchico a 4 livelli: 3 funzionali ed 1 tecnico**;
- sull'utilizzo di **TFS come strumento** per la gestione e raccolta dei requisiti.

5.1 CLASSIFICAZIONE DEI REQUISITI

Il modello selezionato prevede la seguente classificazione dei requisiti:

- **Theme**: è il più alto livello della gerarchia. Fornisce una visione del prodotto finale. È suddiviso in una o più epiche che contribuiscono ad uno stesso obiettivo,
- **Epic**: Organizza il lavoro necessario a completare una parte del *Theme* suddividendolo in componenti più elementari. È una richiesta troppo grande per essere realizzata in uno sprint.
- **Feature**: È un elemento funzionale distinto che garantisce determinate competenze. Sono un insieme di User Story.
- **User Story**: Descrizione della funzionalità desiderata dall'utente/sistema a partire dalla Feature descritta: Natura; storie funzionali.
- **Task**: Parti necessarie a portare a termine una User Story. Sono elementi tecnici e logiche di dettaglio, quali algoritmi di calcolo, strutture dati, ecc; Consentono di stimare esattamente il tempo necessario al completamento di una Storia.

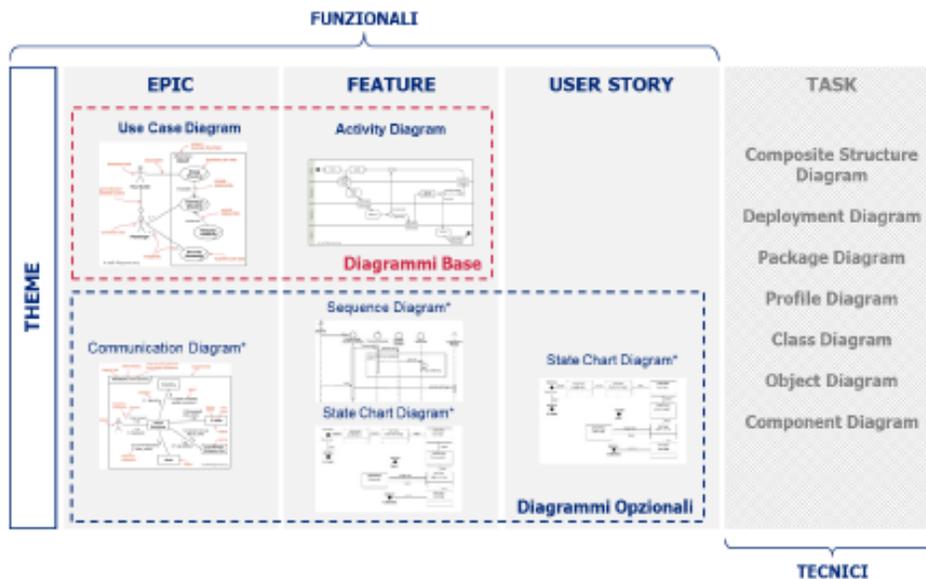
Per ciascuna classe sono definite le responsabilità assunte nella definizione dei requisiti dai team progettuali Modello di servizi e Sviluppo sw.



Il team progettuale Modello di Servizio oltre a dettagliare le feature/epic identificate in fase di definizione e raccolta requisiti, sarà responsabile di validare/aggiornare eventuali feature proposte/aggiornate dall'Uff. VIII e Uff. IX;

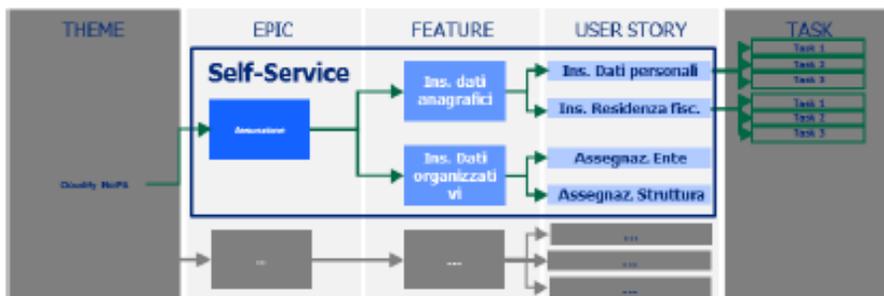
Report

A supporto della definizione e condivisione dei requisiti si è deciso di utilizzare come **strumento di rappresentazione i Diagrammi UML**, in particolare gli User Case Diagram e gli Activity Diagram, oltre che ulteriori Diagrammi opzionali (Communication Diagram; Sequence Diagram, State Chart Diagram).



Formalmente per ogni epica deve essere definito un documento con l'obiettivo di:

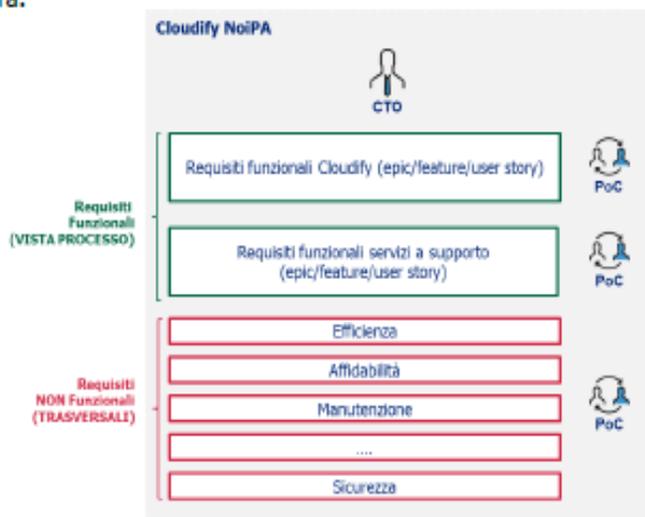
- fornire un template e le linee guida condivise per la stesura dei requisiti in attesa dell'adeguamento di TFS;
- fornire una possibile struttura di riferimento per la generazione della documentazione dei requisiti mediante TFS (da verificare fattibilità tecnica);
- contenere, a partire da una singola epica, tutte le Features con le relative User Stories.



Si riepilogano di seguito le tipologie e le caratteristiche dei requisiti da gestire del programma Cloudify:

- I **requisiti funzionali** Cloudify NoiPA saranno caratterizzati per processo (epic/feature) dal gruppo modello di servizio e (user story) dal gruppo sviluppo sw
- I **requisiti funzionali dei servizi a supporto** saranno caratterizzati per processo (epic/feature/user story) da diversi gruppi di lavoro
- I **requisiti non funzionali trasversali** saranno definiti in un numero contenuto dal gruppo

infrastruttura.



Sono state identificate alcune aree tematiche che costituiscono i servizi a supporto di Cloudify:

- *logiche applicative;*
- *gestione ruoli e utenti;*
- *cruscotto gestionale processi;*
- *monitoraggio applicativo;*
- *monitoraggio infrastrutturale; monitoraggio dati;*
- *formazione;*
- *knowledge management;*
- *assistenza*

Con riferimento alla definizione dei requisiti per tali servizi, si evidenzia che:

- Per i servizi di Formazione, Knowledge management e assistenza saranno definiti dal gruppo **Comunicazione**.
- Per i servizi di Monitoraggio, Cruscotto Gestionale, Gestione Ruoli e Utenti, Logiche Applicative saranno definiti dal gruppo di **Sviluppo SW**.
- L'attività di monitoraggio sarà effettuata separatamente sui tre stream progettuali (Infrastrutture, Sviluppo SW, Data Gov.)

5.2 MODELLO DI GOVERNANCE

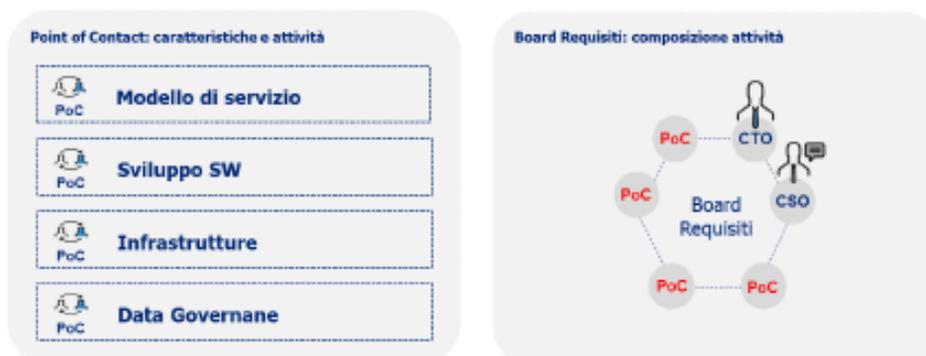
Il modello di governance del processo di gestione dei requisiti prevede la costituzione di un board dei requisiti costituito da CTO e PoC (Point of Contact) rappresentanti delle seguenti 4 linee progettuali (Modello di servizio, Infrastrutture, Sviluppo SW, Data Governance).

5.2.1 POINT OF CONTACT: CARATTERISTICHE E ATTIVITÀ

I PoC identificati dai gruppi di lavoro dovranno avere una buona conoscenza dei requisiti implementati dall'attuale sistema NoiPA.

Durante tutto il processo di raccolta dei requisiti dovranno garantire la validazione dei requisiti in tempi brevi. Dovranno, inoltre, garantire una vista di insieme su tutti gli stream progettuali.

Il Board requisiti sarà composto dai PoC identificati e guidato dal CTO con il supporto del CSO. Si occuperà della validazione dei requisiti in termini di coerenza e consistenza e avrà il compito di passare i requisiti elaborati alla fase di sviluppo software.



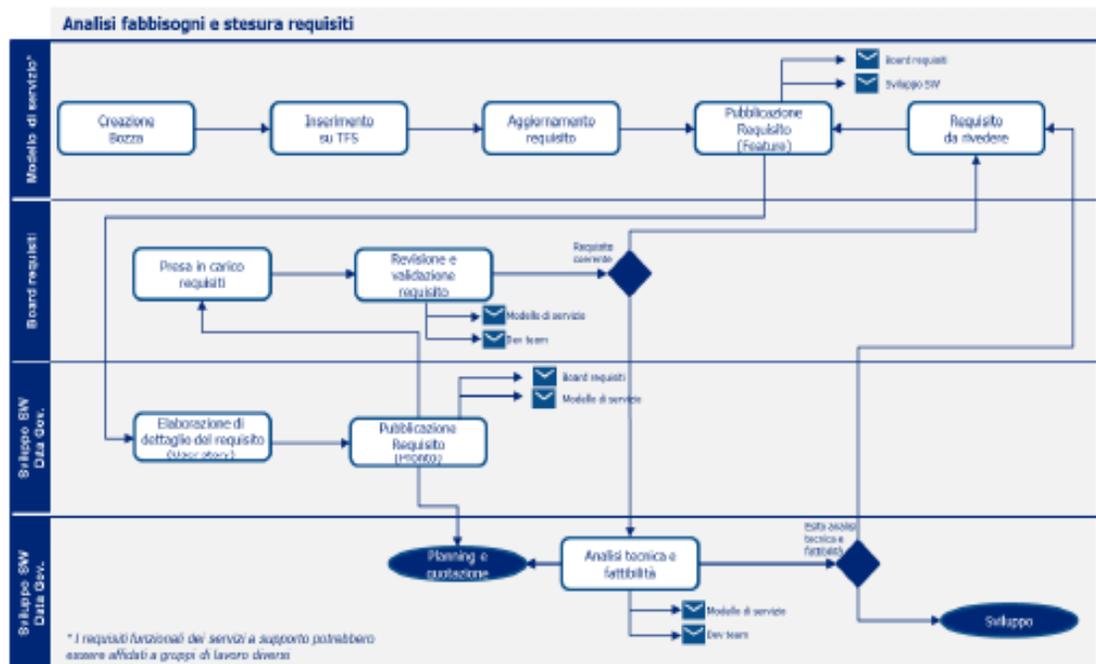
5.3 PROCESSI DI GESTIONE DEI REQUISITI

Il ruolo delle diverse linee progettuali nell'ambito del processo di gestione dei requisiti è:

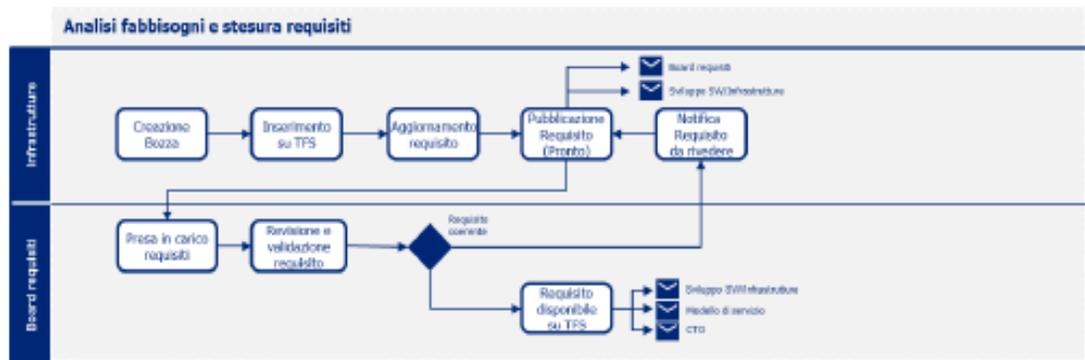
- **Sviluppo Software:** Analizza i requisiti e crea le user story e ne aggiorna lo stato.
- **Modello di servizio:** Crea i requisiti a livello di epic e feature e ne aggiorna lo stato.
- **Board requisiti:** Esegue una verifica sui requisiti. Aggiorna lo stato dei requisiti passandoli allo stato «in analisi». Ha diritti di lettura e scrittura su tutti gli stati.
- **Infrastruttura:** Crea i requisiti non funzionali e ne aggiorna lo stato. Ha diritti di lettura e scrittura dei requisiti negli stati: bozza, pronto. Ha diritti solamente di lettura negli altri stati.

5.3.1 *PROCESSO GESTIONE REQUISITI FUNZIONALI*

Si riporta di seguito il processo di gestione dei requisiti funzionali, con l'evidenza delle principali attività che i soggetti coinvolti devono effettuare e delle dipendenze tra di esse.



5.3.2 *PROCESSO GESTIONE REQUISITI NON FUNZIONALI (TRASVERSALI)*



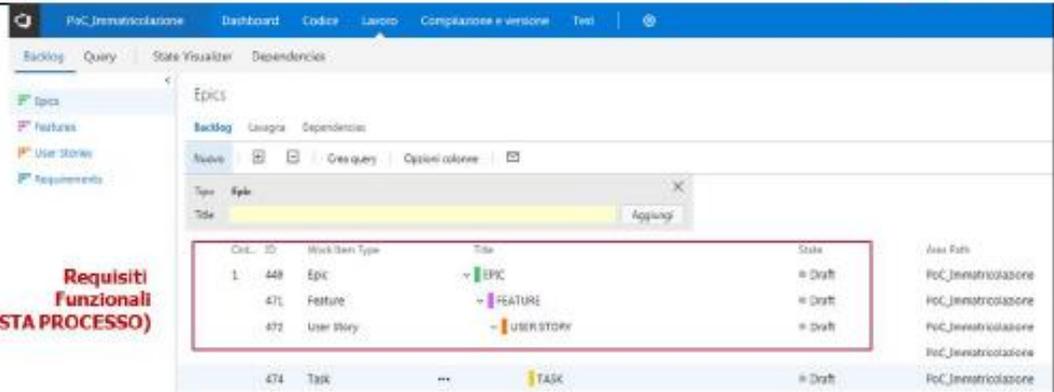
5.4 TFS E LA GESTIONE DEI REQUISITI

Lo strumento identificato per la gestione dei requisiti Cloudify è il TFS.

La tassonomia per i requisiti funzionali su TFS sarà composta dai seguenti livelli: Epica, Feature e UserStory.

Inoltre ogni elemento consentirà la ricerca full-text, rendendo veloce il reperimento delle informazioni.

Report



Requisiti Funzionali (VISTA PROCESSO)

Ord.	ID	Work Item Type	Title	State	Area Path
1.	448	Epic	EPIC	Draft	PoC_Innistricolazione
	471	Feature	FEATURE	Draft	PoC_Innistricolazione
	472	User Story	USER STORY	Draft	PoC_Innistricolazione
	474	Task	TASK	Draft	PoC_Innistricolazione

Il collegamento delle user story riutilizzate in più epiche e feature avverrà tramite riferimenti nel sistema TFS, al fine di evitare la duplicazione della stessa US in più punti del sistema;

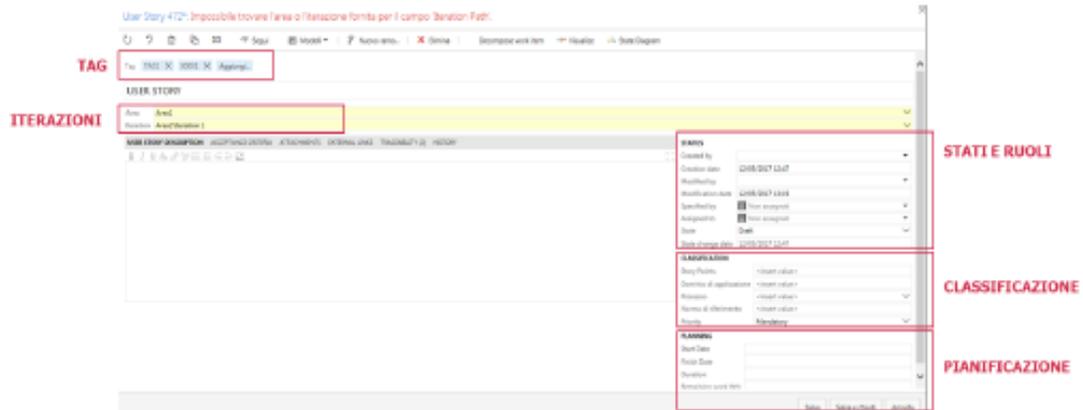
I requisiti saranno collegati ai pacchetti di servizi tramite tag nel sistema TFS (es. L'epica Assunzione avrà i tag Built in, G1, P1, E1).

Livello	Regola	Applicazione per CloudifyNoiPA	Note
Tema	T + prime 3 consonanti	TCLD	In caso di omocodia: la terza consonante viene sostituita da un numero progressivo
Epica	E + prime tre consonanti	ESSN – Epica ASSUNZIONE	In caso di omocodia: la terza consonante viene sostituita da un numero progressivo
Feature	F + acronimo formato dalle iniziali delle prime 4 parole della descrizione estesa della feature	FIDAS – Feature Inserimento dati anagrafici soggetto	nel caso in cui non ci siano 4 parole, progressivo numerico partendo da 0 a completamento dei 4 caratteri (es Ricerca amministrato FRA00)
User Story	U + progressivo numerico univoco (0-999999)	E0001 - Inserimento dati personali	Questo consente il collegamento delle User story a più feature in maniera agevole

Ogni elemento della tassonomia (es. User Story) potrà essere caratterizzato su diverse dimensioni:

- **Tag:** ogni elemento potrà essere etichettato trasversalmente (es. per catalogo servizi, per stakeholder, etc.)
- **Iterazioni:** Ogni elemento potrà essere agganciato ad un'area di lavoro ed una iterazione (per gestire sprint gli sprint in parallelo)
- **Stati e Ruoli:** TFS consente di definire degli stati (coerenti con il processo ed i ruoli associati ai diversi passaggi di stato)
- **Classificazione:** si possono definire diverse categorie con cui classificare i requisiti (es. norma di riferimento)

- **Pianificazione:** Ogni elemento potrà essere pianificato nelle varie transizioni di stato.



Per ogni user story sarà possibile definire dei campi che la classificano, utili anche per la successiva ricerca.

I campi per la classificazione potranno essere «liberi» (campo testo) oppure «selezionabili» (menù a tendina).

Tipo	Descrizione	Valori
Selezionabile	Catalogo servizi	Built-in; Giuridico; Economico; Presenze;
Selezionabile	Priorità	da 0 a 5 (0 Poco Prioritario – 5 Prioritario)
Selezionabile	Applicabilità	Lista comparti;
Selezionabile	Provenienza	Comparto; Ente; Prospect; Normativa;
Libero	Norma di riferimento	Campo libero

I requisiti non funzionali (trasversali) potranno essere gestiti senza la tassonomia ma con una opportuna classificazione, di seguito una proposta:

- accessibilità
- affidabilità
- disponibilità
- efficienza
- interoperabilità
- normativo
- performance
- portabilità
- privacy
- riuso
- usabilità
- qualità
- manutenzione
- sicurezza

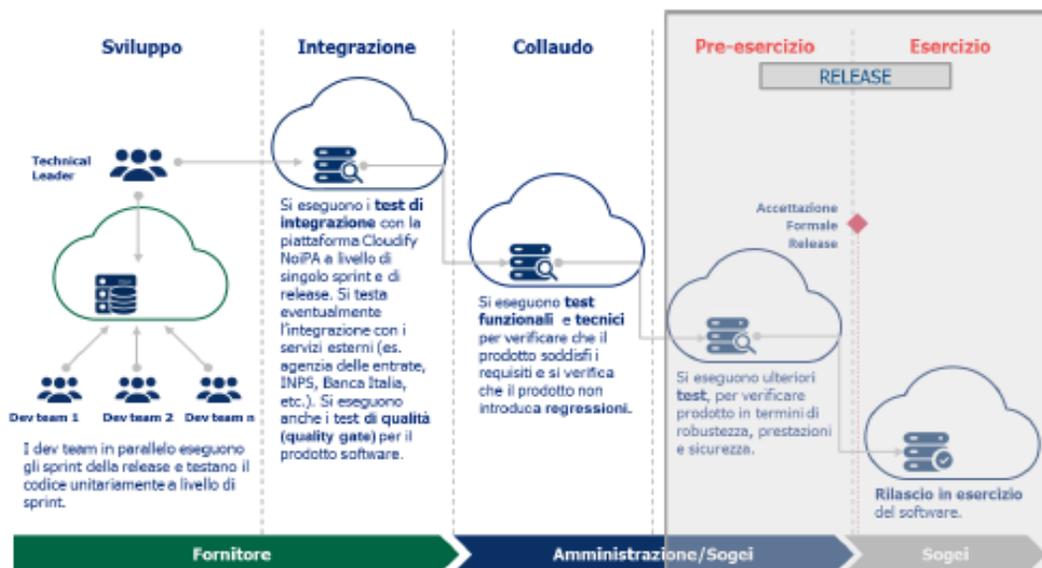
Per ogni requisito dovrà essere possibile definire l'appartenenza a più categorie (ad es. tramite una checkbox).

6. IL PROCESSO DALLO SVILUPPO ALL'ESERCIZIO: AMBIENTI, RUOLI E RESPONSABILITÀ

6.1 GLI AMBIENTI

Gli ambienti identificati per lo sviluppo sw in modalità Agile sono 5. Di seguito si riporta il dettaglio di ciascuno di essi.

- Attività eseguita sull'ambiente dal Fornitore
- Attività eseguita sull'ambiente da Amministrazione/Sogei
- Attività eseguita sull'ambiente Sogei



Nota: è previsto anche un ambiente di pre-integrazione, messo a disposizione dall'Amministrazione per favorire l'attività di test da parte del fornitore, che non fa parte integrante del processo.

6.2 I TEST

Gli output dei diversi sprint e delle release saranno sottoposti ad una serie di verifiche. Di seguito le tipologie di test da implementare:

- **Test Integrazione:** verifica del software sviluppato dal fornitore su proprio ambiente (SVILUPPO) nell'ambito della piattaforma messa a disposizione del MEF (prima verifica su ambiente di INTEGRAZIONE). I test di integrazione, (oltre a prevedere l'esecuzione di tutti gli unit-test) prevedranno test ad-hoc sul dialogo tra/con sistemi interni ed esterni.
- **Test di Qualità SW:** necessario identificare benchmark e policy condivise al fine di stabilire dei quality gate sulla qualità del prodotto software;
- **Test Funzionali:** verifica funzionale eseguita a livello di singolo requisito (user story) rispetto a: definition of done e criteri di accettazione definiti;
- **Test di Performance:** necessario identificare benchmark e policy condivise al fine di stabilire i livelli di performance attesi;

- **Test di Security:** rispetto ai requisiti di sicurezza fino ad ora identificati (Eg. OWASP) è necessario identificare i requisiti tipici del sistema Cloudify NoiPA (Eg. mascheramento dati sensibili, definizione di cono di visibilità, etc.);

6.2.1 GLOSSARIO DEI TEST

Ai fini di maggiore chiarezza si riporta di seguito un breve glossario con la descrizione dei test e la collocazione degli stessi negli specifici ambienti, indicando se il test avviene in automatico (A) o manualmente (M):

- **(A) - Unit test:** test unitari del codice eseguiti in fase di compile/build del codice;
- **(A) - Integration/Component test:** test di integrazione e di componente eseguiti su intero stack software rilasciato sia a livello di release che di singolo sprint. I test, oltre a verificare la corretta integrazione del software sviluppato in parallelo tra i diversi DevTeam, dovranno verificare la corretta integrazione con sistemi interni ed esterni;
- **(A) - Quality test:** verifica statica della qualità del codice secondo check list e KPI definite e configurate sui diversi strumenti;
- **(A) - Performance test:** verifica puntuale su componenti architetturali core della piattaforma attraverso l'uso di check list e KPI definite e configurate sui diversi strumenti;
- **(A) - Smoke test:** verifica su sotto-insieme rilevante per la release sia in termini funzionali che di integrazione al fine di garantire la non regressione della nuova release software;
- **(A) - Post-deploy test:** test su componenti applicative rilasciate sui diversi ambienti;
- **(M) - Visual & Usability Test:** test su UX e UI;
- **(M) - User Acceptant test:** test di accettazione funzionale secondo criteri di accettazione e test-case definiti;
- **(A) - Stress test:** stress test per verificare scalabilità, stabilità e robustezza del pacchetto software candidato al rilascio su ambiente di produzione;
- **(A) - Penetration test:** Il test ha come obiettivo evidenziare le debolezze della piattaforma fornendo il maggior numero di informazioni sulle vulnerabilità che ne hanno permesso l'accesso non autorizzato. L'analisi è condotta dal punto di vista di un potenziale attaccante;
- **(A) - Ongoing live transaction test:** test su componenti/elementi transazionali core della piattaforma prima dell'apertura all'utenza;

	PoC	Integrazione	Collaudo	Pre-esercizio	Esercizio
Integration/Component test	SI	SI	NO	NO	NO
Quality test	SI	SI	NO	NO	NO
Unit test	SI	SI	NO	NO	NO
Performance test	SI	NO	SI	NO	NO
Stress test & Penetration Test	NO	NO	NO	SI	NO
Security test	NO	NO	SI	NO	NO
Smoke test	NO	NO	NO	SI	NO
Post-deploy test	NO	NO	SI	SI	SI
Visual & Usability Test	SI	NO	SI	NO	NO
User Acceptant test	NO	NO	SI	NO	NO
Ongoing live transaction test	NO	NO	NO	NO	SI

Nota: I test da realizzare fanno riferimento nel dettaglio al Piano di Qualità Cloudify predisposto da Sogei

Report

6.2.2 I TOOL

Per ciascuna fase del processo di sviluppo Agile è stata definita una prima ipotesi di impiego dei tool applicativi disponibili. Tali strumenti saranno verificati ed integrati con altri strumenti disponibili.

6.2.3 RUOLI E RESPONSABILITÀ

Nella RACI seguente si riporta dettaglio dei ruoli e delle responsabilità definite per le attività di **rilascio degli sviluppi a livello di sprint e release in ambiente di integrazione, collaudo, pre-esercizio ed esercizio.**

	PMO	Modello di servizio	Comunicazione	Sviluppo Software	Infrastrutture	Governance Data	CTO	Linea Guida Qualità SW	PO	PMO	SM	DevTeam
Rilascio in ambiente di integrazione					C				I	A	R	R
Sprint Review - Test di integrazione e di qualità										A	R	R
Release Review - Test di qualità (opzionali)										A	R	R
Release Review - Test di integrazione (ambiente integrazione)										A	R	R
Accettazione test integrazione e qualità					R	R	C		R	A/R	C	C
Rilascio in ambiente di collaudo					C				A/R	R		R
Sprint Review - Test funzionali e di non regressione (ambiente di collaudo)									A/R	R		R
Release Review - Test funzionali e di non regressione (ambiente di collaudo)		C		A/R	I	C						
Accettazione test funzionali e di non regressione (su Sprint)					R	R	C		C	A/R		C
Accettazione test funzionali e di non regressione (su Release)		R		A/R	R	R	C					
Rilascio in ambiente di pre-esercizio			I	A/R	C							
Release Review - Test di performance e security (ambiente di pre-esercizio)				R	A/R	R						R
Accettazione test di performance e security					A/R							
Accettazione (a valle di superamento di tutti i test).			I	A/R			C					
Rilascio in ambiente di esercizio			I		R		A					

Descrizione:

Ambiente dedicato alle attività di sperimentazione delle PoC sui per quanto concerne il team di Sviluppo Software sia per quanto concerne il team di Data Governance.

Sistemi

Interni/Esterni integrati: NO

Integrato in Processo di CI/CD: NO

Stack Tecnologico

Presentation Platform (WS)	SI
Api Manager	SI
Portal & CMS	NO
Service Platform (AS)	SI
Rule Engine	SI
Business Process Management	SI
Enterprise Service Bus	SI
Data Virtualization	SI
Master Data Management	SI
RDBMS	SI
IAM (OAM)	NO

Testing

Integration/Component test	SI
Quality test	SI
Unit test	SI
Performance test	SI
Stress test	NO
Penetration Test	NO
Security test	NO
Smoke test	NO
Post-deploy test	NO
Visual & Usability Test	SI
User Acceptant test	NO
Ongoing live transaction test	NO

6.2.5 *AMBIENTE INTEGRAZIONE*

Descrizione: Ambiente a disposizione dei team di sviluppo per garantire la corretta integrazione su ambiente MEF delle componenti applicative sviluppate cross-team durante la fase di Sprint-Execution

Stack Tecnologico

Presentation Platform (WS)	SI
Api Manager	SI
Portal & CMS	SI
Service Platform (AS)	SI
Rule Engine	SI
Business Process Management	SI
Enterprise Service Bus	SI
Data Virtualization	SI
Master Data Management	SI
RDBMS	SI
IAM (OAM)	SI

Testing

Integration/Component test	SI
Quality test	SI*
Unit test	SI
Performance test	NO
Stress test	NO
Penetration Test	NO
Security test	NO
Smoke test	NO
Post-deploy test	NO
Visual & Usability Test	NO
User Acceptant test	NO
Ongoing live transaction test	NO

*(Opzionale)

Sistemi Interni/Esterni integrati: SI (da definire elenco mock-interface)

Integrato in Processo di CI/CD: SI – Passaggio automatico su ambiente di collaudo a valle del superamento dei test.

Descrizione: Ambiente deputato al collaudo tecnico e funzionale. E' a disposizione dei product owner per verificare la corrispondenza con i requisiti funzionali e di usabilità definiti durante la fase di sprint-review e/o release review

Stack Tecnologico		Testing	
Presentation Platform (WS)	SI	Integration/Component test	NO
Api Manager	SI	Quality test	SI
Portal & CMS	SI	Unit test	NO
Service Platform (AS)	SI	Performance test	SI
Rule Engine	SI	Stress test	NO
Business Process Management	SI	Penetration Test	NO
Enterprise Service Bus	SI	Security test	SI
Data Virtualization	SI	Smoke test	NO
Master Data Management	SI	Post-deploy test	SI
RDBMS	SI	Visual & Usability Test	SI
IAM (OAM)	SI	User Acceptant test	SI
		Ongoing live transaction test	NO

Sistemi Interni/Esterni integrati: SI (da definire elenco mock-interface)

Integrato in Processo di CI/CD: SI – Passaggio semi-automatico su ambiente di pre-produzione a valle del superamento dei test

Descrizione: Ambiente dedicato all'esecuzione di test long-run su intero stack tecnologico ed applicativo (eventualmente utilizzato per preview su utenza amica) durante la fase di release-review.

Stack Tecnologico		Testing	
Presentation Platform (WS)	SI	Integration/Component test	NO
Api Manager	SI	Quality test	NO
Portal & CMS	SI	Unit test	NO
Service Platform (AS)	SI	Performance test	SI*
Rule Engine	SI	Stress test	SI
Business Process Management	SI	Penetration Test	SI
Enterprise Service Bus	SI	Security test	NO
Data Virtualization	SI	Smoke test	SI
Master Data Management	SI	Post-deploy test	SI
RDBMS	SI	Visual & Usability Test	NO
IAM (OAM)	SI	User Acceptant test	NO
		Ongoing live transaction test	NO

* (Opzionale)

Sistemi Interni/Esterni integrati:SI (da definire elenco mock-interface e sistemi con interfacce di test)

Integrato in Processo di CI/CD:SI – Passaggio semi-automatico su ambiente di pre-produzione a valle del superamento dei test

6.2.8 *AMBIENTE ESERCIZIO*

Descrizione:
Ambiente di produzione;

Sistemi Interni/Esterni integrati: SI

Integrato in Processo di CI/CD: SI

Stack Tecnologico		Testing	
Presentation Platform (WS)	SI	Integration/Component test	NO
Api Manager	SI	Quality test	NO
Portal & CMS	SI	Unit test	NO
Service Platform (AS)	SI	Performance test	NO
Rule Engine	SI	Stress test	NO
Business Process Management	SI	Penetration Test	NO
Enterprise Service Bus	SI	Security test	NO
Data Virtualization	SI	Smoke test	NO
Master Data Management	SI	Post-deploy test	SI
RDBMS	SI	Visual & Usability Test	NO
IAM (OAM)	SI	User Acceptant test	NO
		Ongoing live transaction test	SI

7. FOCUS SPRINT REVIEW

7.1 SPRINT REVIEW: RUOLI, PROCESSO E TEMPISTICHE

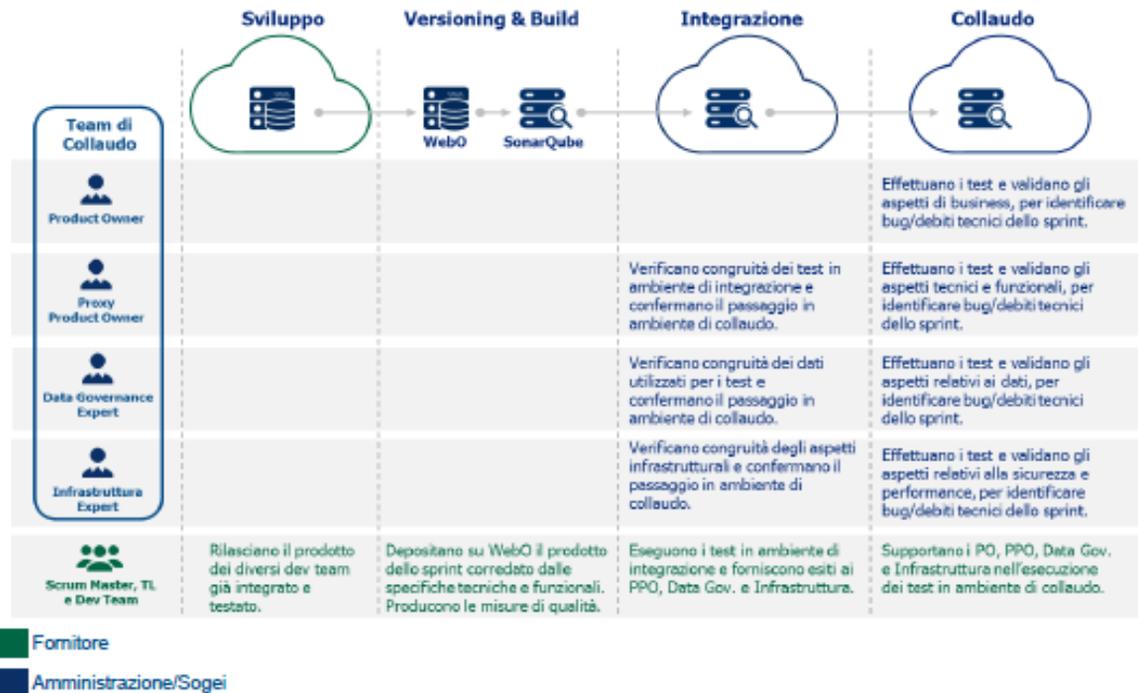
Il presente capitolo è una guida per uno svolgimento ordinato ed efficiente del processo di verifica degli Sprint del progetto Cloudify, che complessivamente viene definito Sprint Review.

In particolare si dettagliano:

- processo, gli attori, i ruoli e le responsabilità
- ambienti di riferimento dei diversi step e le tipologie di test che vengono effettuati in ciascuno di essi
- modalità di conduzione dei collaudi
- tempistiche di riferimento per ciascuno step
- template a supporto della fase di sprint review

Nello schema seguente si evidenziamo le attività, relative alla fase di sprint review, svolte:

- da SM, TL e Dev Team negli ambienti di Sviluppo, Integrazione e Collaudo
- dal Team di collaudo con il dettaglio delle verifiche/test effettuate da PO, PPO, Data Gov Expert e Infrastruttura Expert negli ambienti di Integrazione e Collaudo



Per ciascuna delle attività eseguite da PPO, Data Gov Expert e Infrastruttura Expert nella fase di **rilascio in collaudo** vengono specificati Input, Strumenti, Output

Input	Attività	Strumenti*	Output
 <ul style="list-style-type: none"> • Esiti unit test • Esiti integration test • Misure di qualità 	<ul style="list-style-type: none"> • Verifica esiti unit test • Verifica esiti integration test • Verifica misure di qualità 	<ul style="list-style-type: none"> • TFS • WebQ/GIT • SonarQube 	<ul style="list-style-type: none"> • Accettazione esiti test • US realizzate e acceptance criteria • Rilascio in ambiente di Collaudo
 <ul style="list-style-type: none"> • Modello fisico dei dati • Dati utilizzati per i test in ambiente di integrazione 	<ul style="list-style-type: none"> • Verifica dati utilizzati per i test in ambiente di integrazione • Verifica corretto utilizzo del modello fisico dei dati 	<ul style="list-style-type: none"> • TFS • Strumenti per l'accesso alle basi dati (da identificare) 	<ul style="list-style-type: none"> • Conformità sui dati utilizzati per i test in ambiente di integrazione
 <ul style="list-style-type: none"> • Configurazione ambiente integrazione • Documentazione deploy in integrazione 	<ul style="list-style-type: none"> • Verifica configurazione ambiente di integrazione • Verifica della documentazione relativa al deploy in integrazione 	<ul style="list-style-type: none"> • WebQ/GIT • SonarQube 	<ul style="list-style-type: none"> • Conformità aspetti infrastrutturali, di configurazione e deploy

Rilascio in ambiente di collaudo

* eventualmente da integrare con ulteriori strumenti

Per ciascuna delle attività eseguite **nella fase collaudo** da PO, PPO, Data Gov Expert e Infrastruttura Expert vengono specificati Input, Strumenti, Output

Input	Attività	Strumenti*	Output
 <ul style="list-style-type: none"> • Definition of Done • Lista US realizzate e acceptance criteria • Piano di test e test case • Esiti test integrazione 	<ul style="list-style-type: none"> • Integrazione e validazione del piano di test • Verifica req. di business • Esecuzione visual e usability test (UX/UI) 	<ul style="list-style-type: none"> • TFS (fino al completamento della configurazione per la gestione dei test case, si dovranno utilizzare appositi template) 	<ul style="list-style-type: none"> • Esito complessivo dello sprint • Esiti visual e usability test
 <ul style="list-style-type: none"> • Lista US realizzate e acceptance criteria • Piano dei test di non reg. • Documentazione tecnica • Misure di qualità 	<ul style="list-style-type: none"> • Esecuzione test funzionali • Test di non regressione • Analisi del codice • Quality gate 	<ul style="list-style-type: none"> • TFS • WebQ/GIT • SonarQube 	<ul style="list-style-type: none"> • Lista US validate e debiti tecnici • Esiti test funzionali e di non regressione • Conformità prodotto SW
 <ul style="list-style-type: none"> • Dati utilizzati per i test in ambiente di collaudo 	<ul style="list-style-type: none"> • Verifica dati utilizzati per i test in ambiente di collaudo • Verifica corretto utilizzo del modello fisico dei dati 	<ul style="list-style-type: none"> • TFS • Strumenti per l'accesso alle basi dati (da identificare) 	<ul style="list-style-type: none"> • Conformità sui dati utilizzati per i test in ambiente di collaudo
 <ul style="list-style-type: none"> • Documentazione deploy in collaudo • Piano di test performance e security 	<ul style="list-style-type: none"> • Verifica della documentazione relativa al deploy in collaudo • Esecuzione dei test di performance e security 	<ul style="list-style-type: none"> • WebQ/GIT • WebInspect • LoadRunner • SilkPerformer 	<ul style="list-style-type: none"> • Conformità aspetti infrastrutturali, di configurazione e deploy • Esiti test di security e performance

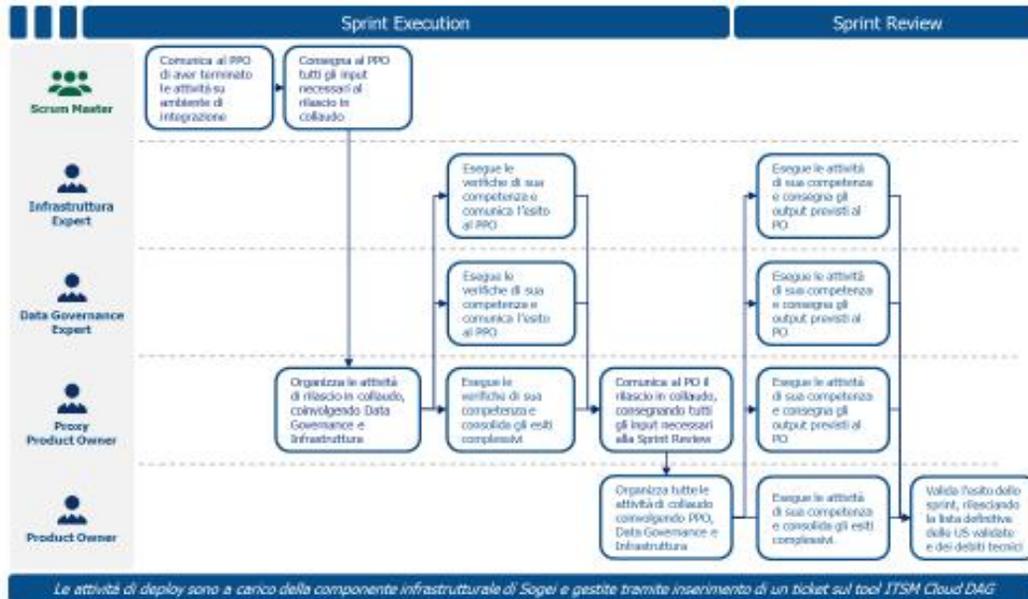
Validazione sprint e lista debiti tecnici/bug

* eventualmente da integrare con ulteriori strumenti

Report

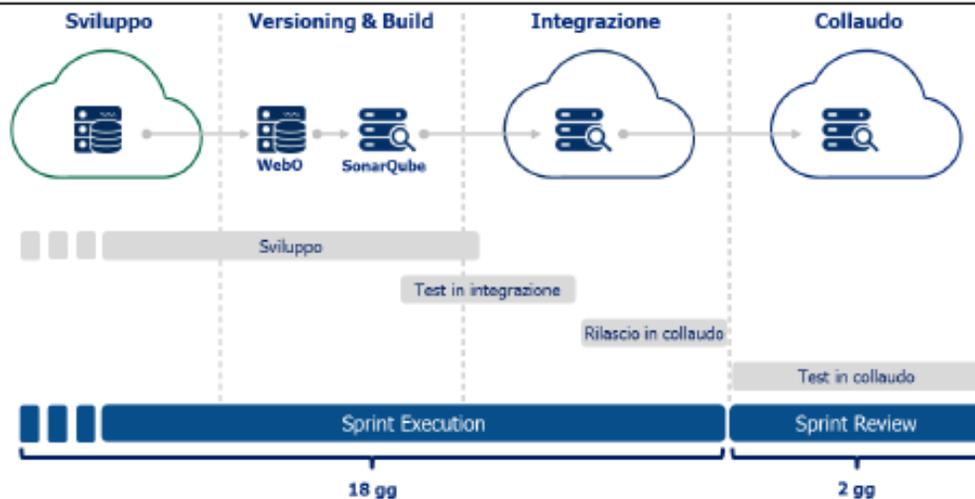
Ai fini di un maggior dettaglio delle attività eseguite nei diversi ambienti da tutti gli attori coinvolti si rimanda al documento del CTO_Definition of done.

Nel seguente processo si evidenziano le attività, con particolare riferimento alle attività di coordinamento effettuate da ciascuno ruolo nell'ambito della sprint execution, in fase di rilascio in collaudo, e nel corso dello sprint review.



La durata di uno sprint è pari a 20 gg di cui 18 gg di execution e 2 gg di sprint review, la Sprint Retrospective avviene entro 5 gg dalla fine del collaudo.

Report



Nota: la Sprint Retrospective avviene entro 5 giorni dalla fine del Collaudo

Entro 5 giorni dalla Sprint Review, viene fatta la **Sprint Retrospective** che ha come obiettivi:

- ✓ **Esaminare** come l'ultimo Sprint è andato in merito a persone, relazioni, processi e strumenti
- ✓ **Identificare e ordinare** gli elementi principali che sono andati bene e le migliorie potenziali
- ✓ **Creare** un piano per attuare i miglioramenti al modo di lavorare dello Scrum Team

Partecipano a tale attività Scrum Master, Proxy Product Owner e Product Owner

La tecnica utilizzata è quella dello Startfish, che consiste nello scrivere su dei post-it le attività dello sprint, suddivise in cinque categorie:

- **Stop** - attività che non portano valore
- **Less** - attività che richiedono molto lavoro e producono pochi benefici
- **Keep** - attività che i membri del team vogliono mantenere, poiché portano benefici
- **More** - attività sulle quali il team dovrebbe focalizzarsi e lavorare più spesso
- **Start** - attività nuove che il team vuole attuare



7.2 TEMPLATE PER SPRINT REVIEW

7.2.1 OBIETTIVI DELLO SPRINT

ID User Story	Titolo User Story	Stato	Note
U000xx	Xxx		
U000xx	Xxx		
			

(*) Le US che al momento della Sprint Review non sono state ancora lavorate o comunque non sono chiuse, vengono classificate come «da ripianificare»

 US chiusa  US chiusa con debito  US da ripianificare

7.2.2 SEGNALAZIONI RISOLTE E DA VERIFICARE NEL CORSO DELLA REVIEW

Sprint	ID User Story	Titolo User Story	Tipo	Descrizione	Note
Sprint 1	U000xxx	xxx	xxx		
	U000xxx	xxx	xxx		

Le segnalazioni possono essere dei seguenti tipi:

- o **Anomalia:** difetto di sviluppo da sanare
- o **Debito tecnico:** funzionalità tecnica o controllo non implementati
- o **Debito funzionale:** funzionalità tecnica o controllo assente, per necessità di

Report

- o **Richiesta:** nuova funzionalità tecnica o controllo non indicato precedentemente nei criteri di accettazione
- o **Interfaccia - richiesta:** nuova richiesta di integrazione UX e UI – da inserire nelle linee guida
- o **Interfaccia - anomalia:** richiesta di integrazione UX e UI indicata ma non ancora recepita
- o Nell’ambito dello Sprint 2 della Release 1 non sono state recepite le segnalazioni di tipo interfaccia (né come risoluzione di segnalazioni dello sprint precedente né come sviluppi dello sprint corrente), lo saranno al termine dello Sprint 3

7.2.3 TRACCIATURA SEGNALAZIONI

Modello di rilevazione delle segnalazioni che emergono durante la Sprint Review

ID	User story	Type	Aperto da	Created	Indicazioni FO	Note SHT/IT
2	U00057 Inserimenti indirizzi istituzionali	Richiesta	Team 2	3-Aug-17	In caso di copia stato estero, oltre a località, deve essere possibile popolare l'indirizzo	Se si seleziona uno stato diverso da Italia attualmente i campi richiesti sono: Stato e Località Estera. Si richiede l'aggiunta del campo Indirizzo (obbligatorio). I campi predisposti di Civico e CAP nel caso di indirizzi esteri non saranno abilitati e l'operatore inserirà nel campo indirizzo anche postal code, civico ed ogni altra informazione che possa ritenere utile per l'estero. Al salvataggio fatto per mancanza di dati si richiede che il sistema indichi i campi che effettivamente non sono stati popolati. Anche visualizzare il popup con il messaggio generico si vorrebbe che il sistema evidenziasse in maniera chiara e accessibile i campi da inserire obbligatoriamente. Da valutare se sufficiente per normativa accessibilità riquadro rosso o in aggiunta mettere simbolo vicino ai campi in errore con un'altra prg segnalazione testuale. Nella realizzazione dei risultati si richiede di inserire anche Località Estera, tra Provincia e Stato. Per indirizzi esteri si richiede quindi di visualizzare, Stato, Località Estera e Indirizzo.
5	U00057 Inserimenti indirizzi istituzionali	Richiesta	Team 2	3-Aug-17		
6	U00116 - Ricerca indirizzi	Richiesta	Team 2	3-Aug-17		

7.2.4 PUNTI DI ATTENZIONE

ID	US di riferimento	Descrizione del punto di attenzione

Report

7.2.5 PROPOSTE

Elenco delle proposte che rappresentano uno spunto di lavoro per la Sprint Retrospective

ID proposta	Descrizione	Stato
1	XXX	
2	XXX	
3		

 Proposta accettata
  Proposta rifiutata
  In valutazione