Processo Creazione ed Evoluzione servizi

Linee guida per la gestione delle revisioni dei deliverables

|  |  |  |  |
| --- | --- | --- | --- |
|  | Unità | Nome | Data |
| Elaborata da: | * **Area Servizi ICT** | * **Umberto Properzi** | * **13-01-2012** | |
| Verificata operativamente da: | * **Resp.le Area Servizi ICT** * **Resp.le Area produzione** | * **Luciano Foti** * **Francesco Milo** | * **08-03-2012** * **09-03-2012** | |
| Verificata per conformità al SGQ da: | * **Gestione Qualità** | * **Monica Marchese** | * **12-03-2012** | |
| Autorizzata da: | * **Resp.le Area Risorse Umane** | * **Antonio Martuscelli** | * **26-03-2012** | |

|  |  |
| --- | --- |
| Data di entrata in vigore | **26 marzo 2012** |

Diffusione del documento

|  |
| --- |
| Lista di Diffusione |
| Tutto il personale della funzione ICT |

|  |
| --- |
| Destinatari delle Linee Guida |
| Tutte il personale della funzione ICT |

Storia delle revisioni:

|  |  |  |
| --- | --- | --- |
| N° Revisione | Data | Motivo |
| 2.0 | 26/03/2012 | introdotta strutturazione repository e consistenza degli archivi |

Indice

[1. Scopo e Ambito di applicazione 4](#_Toc320524401)

[2. Definizioni 4](#_Toc320524402)

[3. Lista degli applicativi 4](#_Toc320524403)

[4. Riferimenti normativi e documenti applicabili 4](#_Toc320524404)

[5. Principi Generali 5](#_Toc320524405)

[5.1. Linee Guida 5](#_Toc320524406)

[5.1.1. Gestione delle revisioni per un deliverable di tipo “documento” 5](#_Toc320524407)

[5.1.2. Nomenclatura, archiviazione documenti e distribuzione dei file contenenti i documenti 6](#_Toc320524408)

[5.1.3. Gestione delle revisioni per un deliverable di tipo “pacchetto SW” 7](#_Toc320524409)

[5.1.4. Consistenza degli archivi 8](#_Toc320524410)

[6. Mappa dei rischi 9](#_Toc320524411)

[7. Matrice delle Responsabilità 9](#_Toc320524412)

[8. Allegati 9](#_Toc320524413)

# Scopo e Ambito di applicazione

Questo documento contiene le linee guida da seguire nella gestione delle revisione dei deliverables di un progetto all’interno del macro processo di Service Creation.

Owner delle Linee Guida per la Gestione delle revisioni è la funzione ICT.

# 

# Definizioni

* **EQS:** Equitalia Servizi
* **Service Creation:** insieme di processi e regole per la creazione/evoluzione di un servizio
* **Deliverable:** “oggetto” risultato di un’attività progettuale, può essere un documento (ad esempio una “specifica tecnica”) oppure un pacchetto SW che implementa tutto o parte di un servizio
* **Pacchetto SW:** insieme di moduli SW che contribuiscono all’implementazione di un servizio, ad esempio tutti i moduli SW nelle diverse tecnologie che cooperano per la realizzazione del“servizio stampa Fermi Amministrativi” (SW di controllo flussi e SW document composition)

# Lista degli applicativi

* Non sono previsti strumenti specifici. Utilizzo di MS Office e di tool di project management

# Riferimenti normativi e documenti applicabili

N/A

# Principi Generali

Un deliverable deve essere univocamente riconoscibile ed indirizzabile sia esso un “documento” a cui è necessario far riferimento (*ad esempio, il progetto ha implementato i requisiti espressi nel “Documento di requisiti versione 3.2*”) sia nel caso di un pacchetto SW che implementa tutto o parte di un servizio (*ad esempio segnalare un’anomalia nel pacchetto SW MioServizio versione 1.2.4*).

## Linee Guida

### Gestione delle revisioni per un deliverable di tipo “documento”

La versione di un documento è espressa con una coppia di numeri del tipo “X.Y”.

Il documento è redatto dagli autori e la prima stesura sarà indicata con versione “0.1”.

Le linee guida per la modifica delle revisioni sono le seguenti:

* **La seconda cifra** “**Y**” sarà incrementata per ogni nuova versione del documento che non è ancora stata formalmente approvata
* **La prima cifra** “**X**” sarà incrementata solo ed esclusivamente a seguito dell’approvazione formale del documento

Esempio:

* La prima stesura redatta dagli autori ha versione 0.1
* Successivamente il documento viene modificato e la nuova versione sarà 0.2, tale versione viene inviata in approvazione.
* Il documento viene respinto durante l’incontro di approvazione in quanto sono necessarie delle modifiche
* Gli autori redigono una nuova versione che sarà la 0.3, questa versione viene di nuovo inviata in approvazione
* Il documento viene approvato ed assume quindi la versione 1.0
* Successivamente viene inviata una richiesta di modifica al progetto, gli autori predispongono quindi una nuova versione del documento individuata dalla versione 1.1 che viene inviata in approvazione
* Il nuovo documento viene approvato ed assume la revisione 2.0

### Nomenclatura, archiviazione documenti e distribuzione dei file contenenti i documenti

I documenti prodotti durante le attività di “*Service Creation*” sono archiviati in formato elettronico “*originale*” (*cioè nei formati della suite Microsoft Office come ad esempio .docx*) sia in formato pdf.

L’archivio da utilizzare sarà quello ufficialmente dichiarato nel “project charter” ed utilizzato da tutti i progetti attivi al momento. L’archivio documentale del progetto dovrà avere la struttura comune definita in allegato ed i documenti sono memorizzati nelle rispettive cartelle come riportato nell’allegato stesso.

Oltre ad indicare la Release si dovrà inserire l’indicazione relativa allo “stato” del servizio, i valori ammessi sono: PRODUZIONE e COLLAUDO. Quando una release viene rilasciata in collaudo si apporrà la dicitura “COLLAUDO”, quando questa release verrà rilasciata in esercizio si apporrà la dicitura “PRODUZIONE” e si eliminerà tale dicitura dalla release precedente che viene sostituita, se applicabile. Si avrà quindi la seguente situazione:

|  |  |
| --- | --- |
| Struttura alla data x | Struttura alla data y (Y>X) |
| 1.0.0  1.0.1  1.1.0 – PRODUZIONE  1.2.0 - COLLAUDO | 1.0.0  1.0.1  1.1.0  1.2.0 - PRODUZIONE |

Ovviamente, alla data Y il collaudo della release 1.2.0 è terminato positivamente e la release è stata “promossa” in esercizio.

Il nome del file sarà organizzato nella seguente struttura XXXXXX\_YYYYY\_ZZZZ\_VVV, dove:

* **XXXXXX** rappresenta il codice documento (ad esempio “MRCRZ001”);
* **YYYYY** rappresenta il “servizio” a cui si fa riferimento scritto senza spazi e separando le diverse parole utilizzando la lettera maiuscola (*ad esempio “Stampa Fermi Amministrativi”*);
* **ZZZZ** rappresenta il “codice progetto”;
* **VVV** rappresenta la versione

Esempio per il documento di “*piano di test/collaudo*”, versione 2.1, relativo al servizio “*stampa Solleciti*” nel progetto “*centralizzazione stampe*” avente come codice “XC3”, i nomi dei file saranno:

* MRCRZ008\_StampaSolleciti\_XC3\_2.1.docx
* MRCRZ008\_StampaSolleciti\_XC3\_2.1.pdf

### Gestione delle revisioni per un deliverable di tipo “pacchetto SW”

La versione di un pacchetto SW è espressa con una tripletta di numeri del tipo “**X.Y.Z**”.

La prima versione rilasciata alla fase di Test sarà indicata con versione “**0.0.1**”.

Le linee guida per la modifica delle revisioni sono le seguenti:

* La versione viene incrementata ad ogni rilascio alle fasi di test e collaudo;
* La prima versione rilasciata in esercizio avrà revisione 1.0.0;
* La prima cifra “**X**” viene incrementata quando vengono inserite una o più funzionalità non presenti nelle versioni precedenti e tali nuove funzionalità hanno impatti sull’interfaccia fornita agli utilizzatori del servizio (*ad esempio tracciati record di scambio con AdR o Enti, nuove maschere per servizi on-line, nuovi WSDL per servizi basati su Web Services, etc*);
* La seconda cifra “**Y**” viene incrementata quando vengono inserite una o più funzionalità non presenti nelle versioni precedenti e tali nuove funzionalità NON hanno impatti sull’interfaccia fornita agli utilizzatori del servizio (ad esempio modifiche sul layout di un documento);
* La terza cifra “**Z**” viene incrementata quando viene rilasciata una MAC o una MAA senza impatti sull’interfaccia fornita agli utilizzatori a meno della risoluzione dell’anomalia

Il Pacchetto SW è generalmente composto da un numero di moduli SW spesso anche realizzati secondo diverse tecnologie ed identificare un certo modulo all’interno dell’insieme che contribuisce al servizio è necessario quanto identificare il pacchetto stesso.

Le linee guida espresse precedentemente sono pertanto valide anche per i singoli moduli SW che compongono il servizio e che costituiranno quindi la “struttura” del servizio stesso, l’unica differenza riguarda il concetto di “*impatto sull’interfaccia*” che in questo caso sarà l’interfaccia tra i diversi moduli (*ad esempio tracciati record di scambio o wsdl di chiamate a web service*).

La “struttura” può anche essere costituita da diversi livelli gerarchici che tendono a raggruppare un certo numero di moduli SW “affini” tra di loro, ad esempio da un punto di vista tecnologico oppure funzionale, la figura seguente illustra un esempio di versionamento e di aggregazione dei moduli e del pacchetto SW:



La struttura e le relative versioni dei moduli sono elencate nei documenti di rilascio dei pacchetti SW

### Consistenza degli archivi

Obiettivo del progetto è il rilascio in esercizio di una determinata release del servizio (ad esempio release 2.1.0 del servizio “Stampa Solleciti”), deve pertanto essere facilmente possibile individuare tutti gli “archivi” relativi a tale release.

Gli “archivi” oggetto di allineamento sono:

* Archivio documentale
* Sistema di archiviazione/gestione dei casi di test (ad esempio QADirector o altri)
* Sistema di archiviazione/gestione delle anomalie (ad esempio Bugzilla o altro)
* Masterplan dei progetti
* Sistemi di change management

Al fine di garantire tale allineamento l’archivio documentale è costruito e gestito come riportato in allegato (ad esempio Stampa Solleciti 2.1.0), i casi di test relativi all’iniziativa sono gestiti/memorizzati nel sistema con il nome della release (ad esempio “StampaSolleciti\_2\_1\_0”, il sistema di gestione delle anomalie riporta il nome della release (ad esempio nome del prodotto/servizio “Stampa Solleciti” e versione “2.1.0”), nel masterplan dei progetti l’iniziativa è censita con il nome della release (ad esempio “Stampa Solleciti 2.1.0”), nei sistemi di change management (ad esempio Harvest/Endevor, etc) viene definita la release con lo stesso nome e versione utilizzato negli altri archivi.

# Mappa dei rischi

N/A

# Matrice delle Responsabilità

N/A

# Allegati

1. Struttura repository documentale  
     
   