Manuale di abilitazione al cloud

italia

11 giu 2024

Indice dei contenuti

1 Premessa **1**

2 1. Il Cloud **3**
2.1 1.1 Cos'è il cloud . . . 3
2.2 1.2 Perché usare il cloud . . . 5
2.2.1 1.2.1 Riduzione dei costi . . . 5
2.2.2 1.2.2 Elasticità reale . . . 6
2.2.3 1.2.3 Facilità degli aggiornamenti . . . 6
2.2.4 1.2.4 Riduzione complessità del supporto . . . 7
2.2.5 1.2.5 Riduzione delle attività manuali a basso valore aggiunto . . . 7
2.2.6 1.2.6 Adeguamento normativo in termini di sicurezza e privacy . . . 7
2.2.7 1.2.7 Miglioramento dei servizi . . . 8

3 2. Come iniziare **11**
3.1 2.1 Roadmap di una migrazione . . . 11

4 3. Assessment servizi e infrastruttura **19**
4.1 3.1 Costruire una mappa degli applicativi e dei servizi attivi . . . 19
4.1.1 3.1.1 Lista degli applicativi . . . 19
4.1.2 3.1.2 Licenze . . . 20
4.1.3 3.1.3 Prioritizzazione degli applicativi . . . 21
4.2 3.2 Scheda di assessment dell'applicativo . . . 24
4.2.1 3.2.1 Aspetti tecnologici . . . 25
4.2.2 3.2.2 Vincoli tecnologici . . . 26
4.2.3 3.2.3 Dati . . . 26
4.2.4 3.2.4 Parti interessate . . . 27
4.2.5 3.2.5 Bisogni . . . 27
4.2.6 3.2.6 Mercato . . . 28
4.3 3.3 Analisi costi-benefici . . . 28
4.3.1 3.3.1 Verifica dei costi attuali dell'infrastruttura . . . 29
4.3.2 3.3.2 Stima dei costi dell'infrastruttura cloud . . . 30
4.3.3 3.3.3 Stima dei costi di migrazione al cloud . . . 30
4.3.4 3.3.4 Stima dei costi post-migrazione . . . 31
4.3.5 3.3.5 Valutazione dei costi rispetto ai benefici tangibili ed intangibili . . . 31
4.4 3.4 Migrazione delle licenze software in cloud . . . 33
4.4.1 3.4.1 Tipologie di licenze cloud . . . 34
4.4.2 3.4.2 Gestione delle licenze per la migrazione al cloud . . . 35

5 4. Pianificare la migrazione **37**
5.1 4.1 Le strategie di migrazione . . . 37
5.1.1 4.1.1 Conservazione o Retain . . . 38
5.1.2 4.1.2 Smantellamento o Retire . . . 39
5.1.3 4.1.3 Sostituzione o Re-purchase . . . 40
5.1.4 4.1.4 Trasferimento di host o Re-host . . . 43
5.1.5 4.1.5 Trasferimento di piattaforma o Re-platform . . . 46
5.1.6 4.1.6 Rifattorizzazione/Creazione di una nuova architettura o Re-architect . . . 49
5.2 4.2 Le competenze necessarie . . . 51
5.2.1 4.2.1 Definizione delle competenze necessarie . . . 52
5.2.2 4.2.2 Mappare le competenze secondo un modello di maturità . . . 53
5.2.3 4.2.3 Strumento per la valutazione delle competenze . . . 54
5.3 4.3 SLA richiesti ai servizi qualificati . . . 55
5.4 4.4 Lock-in . . . 56
5.4.1 4.4.1 Tipologie di lock-in nella pubblica amministrazione . . . 57
5.4.2 4.4.2 Come mitigare il lock-in . . . 57

6 5. Eseguire la migrazione: gli applicativi **61**
6.1 5.1 Metodologia di lavoro . . . 62
6.1.1 5.1.1 Team cross-funzionale . . . 62
6.1.2 5.1.2 Iterattività e incrementalità . . . 62
6.1.3 5.1.3 DevOps . . . 63
6.1.4 5.1.4 Collaborazione e confronto continuo . . . 63
6.1.5 5.1.5 Miglioramento continuo . . . 64
6.2 5.2 La preparazione . . . 64
6.3 5.3 Buone pratiche . . . 65
6.3.1 5.3.1 Scalabilità . . . 65
6.3.2 5.3.2 Disponibilità . . . 66
6.3.3 5.3.3 Resilienza . . . 66
6.3.4 5.3.4 Sicurezza . . . 67
6.3.5 5.3.5 Data Privacy . . . 68
6.3.6 5.3.6 Autenticazione ed autorizzazione . . . 68
6.3.7 5.3.7 Interoperabilità . . . 69
6.3.8 5.3.8 Monitoraggio e alerting . . . 69
6.3.9 5.3.9 Automazione . . . 70
6.3.10 5.3.10 Disaster recovery . . . 72
6.3.11 5.3.11 Backup . . . 73
6.4 5.4 Gli scenari di migrazione . . . 74
6.4.1 5.4.1 Virtualizzazione . . . 74
6.4.2 5.4.2 Containerizzazione . . . 75
6.4.3 5.4.3 Ristrutturare l'applicativo . . . 76
6.5 5.5 La validazione . . . 78
6.5.1 5.5.1 Tipologie di testing . . . 78
6.5.2 5.5.2 Validazione delle funzionalità . . . 79
6.5.3 5.5.3 Validazione delle prestazioni . . . 79
6.5.4 5.5.4 Validazione della sicurezza . . . 80

7 6. Eseguire la migrazione: i dati **83**
7.1 6.1 La preparazione . . . 84
7.2 6.2 Buone pratiche . . . 84
7.3 6.3 Gli scenari di migrazione . . . 85
7.3.1 6.3.1 Migrazione verso lo stesso sistema di gestione delle basi dati . . . 85
7.3.2 6.3.2 Migrazione verso una versione più recente del sistema di gestione delle basi dati . . . 85
7.3.3 6.3.3 Migrazione verso un diverso sistema di gestione delle basi dati . . . 85

7.3.4 6.3.4 Migrazione verso un diverso applicativo 86
7.4 6.4 La validazione 86
7.4.1 6.4.1 Test di completezza 86
7.4.2 6.4.2 Appearance test 87
7.4.3 6.4.3 Test di integrazione 87

8 7. Dopo la migrazione **89**
8.1 7.1 Verifica degli indicatori di performance 89
8.1.1 7.1.1 Indicatori di risultato 90
8.1.2 7.1.2 Indicatori di impatto 91
8.2 7.2 Monitoraggio di nuove soluzioni SaaS aggiunte al Cloud Marketplace 92
8.3 7.3 Segnalazione di violazioni di SLA da parte dei fornitori qualificati 93
8.4 7.4 Condividere l'esperienza di migrazione al cloud 93

CAPITOLO 1

Premessa

La strategia Cloud della PA[1] è nata per favorire l'adozione del modello cloud computing nelle pubbliche amministrazioni italiane, in linea con le indicazioni del piano Italia Digitale 2026 nell'ambito del Piano Triennale per l'informatica pubblica 2019-2021[2] e del Piano triennale per l'informatica nella Pubblica Amministrazione[3] emanato da AgID e con le migliori pratiche nel resto dei principali paesi europei e del mondo.

Nella definizione della strategia Cloud delle PA sono stati individuati tre elementi principali che caratterizzano questo percorso di trasformazione:

- **il principio Cloud First**: secondo il quale le PA devono, in via prioritaria, adottare il paradigma cloud (in particolare i servizi SaaS) prima di qualsiasi altra opzione tecnologica per la definizione di nuovi progetti e per la progettazione dei nuovi servizi nell'ambito di nuove iniziative da avviare
- il **modello Cloud della PA**: il modello strategico che si compone di infrastrutture e servizi qualificati da ACN sulla base di un insieme di requisiti volti a garantire elevati standard di qualità per la PA
- Il **Cloud Enablement Program**: il programma che abilita un'organizzazione a migrare il proprio patrimonio IT esistente (infrastrutture e applicazioni) utilizzando infrastrutture e servizi cloud all'interno del modello Cloud della PA. Nell'ambito del programma è stato definito un framework, costituito dall'insieme di unità organizzative (unità di controllo, unità di esecuzione e centri di competenza), risorse, strategie operative e il Cloud Enablement Kit (metodologie, buone pratiche e strumenti) necessari per attuare il Cloud Enablement Program delle PA.

Questa documentazione rientra nell'ambito del Cloud Enablement Program e, in particolare, del Cloud Enablement Kit che contiene l' insieme di metodi, strumenti e buone pratiche che le pubbliche amministrazioni possono usare per la migrazione al cloud di infrastrutture e applicativi esistenti. Data la sua funzione strumentale all'abilitazione al cloud, questa documentazione costituisce l'elemento principale del **Cloud Enablement Kit**.

Il Cloud Enablement Kit è rivolto non solo a tecnici o esperti dell'IT e ai responsabili della trasformazione digitale (RTD[4]), ma anche a chiunque nella pubblica amministrazione sia coinvolto nella gestione di servizi esistenti e/o nella definizione e progettazione di nuovi servizi.

[1] https://cloud.italia.it/strategia-cloud-pa/
[2] https://pianotriennale-ict.italia.it/
[3] https://www.agid.gov.it/it/agenzia/piano-triennale
[4] https://docs.italia.it/italia/piano-triennale-ict/codice-amministrazione-digitale-docs/it/v2017-12-13/_rst/capo1_sezione3_art17.html

Il contenuto di questo documento è strutturato in maniera lineare per accompagnare le pubbliche amministrazioni nell'intero percorso che va dall'identificazione degli applicativi da migrare fino alla valutazione degli indicatori di risultato a migrazione avvenuta. Tuttavia, i capitoli sono strutturati per essere consultati anche indipendentemente a seconda della fase in cui si trova e dal bisogno specifico che l'amministrazione ha.

I macroargomenti trattati di seguito nel Cloud Enablement Kit sono:

- **Il Cloud** (Capitolo 1): introduzione al concetto di cloud e ai suoi vantaggi
- **Come iniziare** (Capitolo 2): quadro generale dei passi da seguire per iniziare con la migrazione al cloud
- **Assessment servizi e infrastruttura** (Capitolo 3): strumenti per mappare e valutare gli applicativi in uso e identificare quelli migrabili con maggiori benefici e minori criticità
- **Pianificare la migrazione** (Capitolo 4): approfondimento sulle strategie di migrazione ed i rispettivi criteri di applicabilità e indicazioni sugli aspetti da tenere in considerazione durante la pianificazione di una migrazione
- **Eseguire la migrazione** (Capitolo 5): consigli sulla metodologia di lavoro e linee guida di carattere tecnico su come eseguire una migrazione a seconda della strategia scelta
- **Migrare i dati** (Capitolo 6): focus sulla migrazione dei dati di un applicativo e sulle buone pratiche da adottare a seconda dello scenario di migrazione in cui ci si trova
- **Dopo la migrazione** (Capitolo 7): valutazioni da effettuare a migrazione avvenuta e indicatori di performance per misurarne il risultato e l'impatto

Ringraziamo le pubbliche amministrazioni che ci hanno accompagnato in questo percorso e senza le quali non sarebbe stato possibile creare il Cloud Enablement Kit. In particolare, i nostri ringraziamenti vanno a: MiBAC, Corte dei Conti, Regione Emilia Romagna, Comune di Milano, Consorzio.IT (società in house del territorio Cremasco) e Comune di Torino.

CAPITOLO 2

1. Il Cloud

L'introduzione del paradigma cloud nelle pubbliche amministrazioni consentirà di ottenere una più alta qualità e sicurezza informatica ad un costo molto minore rispetto alle soluzioni on-premise, aumentando notevolmente l'affidabilità delle infrastrutture IT e facilitando così il rinnovamento complessivo dei servizi IT nel paese.

Alla luce di ciò, la strategia di Cloud Enablement delineata dal Dipartimento per la trasformazione digitale (DTD) e l'Agenzia per l'Italia Digitale (AgID) è volta ad abilitare la progressiva migrazione delle infrastrutture e degli applicativi esistenti verso il cloud e questo documento, il Cloud Enablement Kit, fornisce un insieme di metodi, strumenti e buone pratiche che le pubbliche amministrazioni possono usare per pianificare e attuare la migrazione.

Prima di iniziare a parlare di migrazione però, in questo capitolo vediamo insieme cos'è il cloud e quali sono i vantaggi che ci si aspetta dalla sua adozione nell'ambito della pubblica amministrazione.

2.1 1.1 Cos'è il cloud

Il *cloud computing* (in italiano *nuvola informatica*), più semplicemente cloud, è un modello di infrastrutture informatiche che consente di disporre, tramite internet, di un insieme di risorse di calcolo (ad es. reti, server, risorse di archiviazione, applicazioni software) che possono essere rapidamente erogate come servizio.
Questo modello consente di semplificare drasticamente la gestione dei sistemi informativi, sia eliminando la gestione relativa ad applicativi fruibili direttamente online, sia trasformando le infrastrutture fisiche in servizi virtuali fruibili in base al consumo di risorse.

Rispetto alle tradizionali soluzioni hardware, il modello cloud consente di:

- usufruire delle applicazioni da qualsiasi dispositivo in qualsiasi luogo tramite l'accesso internet
- avere importanti vantaggi di costo nell'utilizzo del software, in quanto consente di pagare le risorse come servizi in base al consumo ("pay per use"), evitando investimenti iniziali nell'infrastruttura e costi legati alle licenze di utilizzo
- ridurre i costi complessivi collegati alla location dei data center (affitti, consumi elettrici, personale non ICT)
- avere maggiore flessibilità nel provare nuovi servizi o apportare modifiche, con costi accessibili

- effettuare in maniera continua gli aggiornamenti dell'infrastruttura e delle applicazioni
- ridurre i rischi legati alla gestione della sicurezza (fisica e logica) delle infrastrutture IT

La maggior parte dei servizi di cloud computing rientra in tre ampie categorie:

- **software-as-a-service** (**SaaS**), ovvero *software come servizio*: si tratta di un metodo per la distribuzione di applicativi software tramite internet, su richiesta e in genere in base a una sottoscrizione. Nel caso di una soluzione SaaS, i provider di servizi cloud ospitano e gestiscono il software e l'infrastruttura sottostante e si occupano delle attività di manutenzione, come gli aggiornamenti. Gli utenti si connettono all'applicazione tramite internet e possono accedervi da diverse tipologie di dispositivi (desktop, mobile, tablet, . . .). A differenza del modello ASP (Application Service Provider[5]), dove i fornitori installano un'istanza di applicazione per ogni cliente personalizzandole a seconda delle richieste di ognuno, il paradigma SaaS fa uso di applicazioni "multi-tenant[6]", cioè noleggiabili da più utenti contemporaneamente, con codice non customizzabile ma uguale per tutti. Un approccio, quest'ultimo, che garantisce il raggiungimento più facile di economie di scala da parte del fornitore.

  Consulta il catalogo dei servizi SaaS qualificati da ACN[7].

- **platform-as-a-service** (**PaaS**)**,** ovvero *piattaforma distribuita come servizio*: si tratta di servizi cloud che forniscono strumenti e ambienti per lo sviluppo, il test, la distribuzione e la gestione di applicazioni software, solitamente tramite strumenti specifici forniti dal provider stesso e pannelli di configurazione fruibili via browser web. Una soluzione PaaS è progettata per consentire agli sviluppatori di progettare e creare concentrandosi sulle funzionalità dell'applicativo, lasciando al fornitore del servizio aspetti come la configurazione, la gestione dell'ambiente di esecuzione dell'archiviazione o dei database.

  Consulta il catalogo dei servizi PaaS qualificati da ACN[8].

- **infrastructure-as-a-service** (**IaaS**), ovvero *infrastruttura distribuita come servizio*: si tratta di servizi cloud che permettono di allocare risorse infrastrutturali fisiche e virtuali (server, macchine virtuali, risorse di archiviazione e networking) su richiesta mediante interfacce grafiche o mediante API (Application Programming Interfaces[9]) con pagamento in base al consumo.

  Consulta il catalogo dei servizi IaaS qualificati da ACN[10].

Vi sono inoltre diversi modelli di dispiegamento dei servizi:

- **cloud pubblico:** offerti da fornitori che mettono a disposizione dei propri utenti/clienti la potenza di calcolo e/o di memorizzazione dei loro data center. Il tipo di servizi cloud che vengono offerti dal fornitore (IaaS, PaaS, SaaS) dipende dalla politica del fornitore stesso, così come il prezzo e la tariffazione. Uno dei maggiori vantaggi del cloud pubblico per il cliente consiste nel fatto che egli può richiedere l'utilizzo dei servizi cloud di cui necessita nel momento in cui effettivamente ne ha bisogno, e solo per il tempo che gli sono necessari. In questo modo, il cliente può ridurre notevolmente gli investimenti in infrastrutture IT e ottimizzare l'utilizzo delle risorse interne che le gestiscono, oltre a trarre vantaggio dai già citati benefici.

- **cloud privato:** installato dall'utente nel suo data center per suo utilizzo esclusivo. I servizi vengono forniti da elaboratori che si trovano interamente nel dominio del cliente che detiene controllo e totale responsabilità della gestione delle macchine sulle quali vengono conservati i dati e vengono eseguiti i suoi processi, assieme ai complessi aspetti relativi alla sicurezza dei dati. Oltre allo scenario in cui si possiede interamente l'infrastruttura sui propri data center, un altro scenario possibile è invece quello in cui l'utente installa il proprio cloud privato nel data center di un terzo soggetto (tipicamente un fornitore di servizi cloud), in cui dispone di macchine dedicate. In questo caso, l'utente ha il controllo delle macchine anche se non risiedono nel suo dominio, e può configurarle secondo le proprie necessità.

[5] https://it.wikipedia.org/wiki/Application_service_provider
[6] https://it.wikipedia.org/wiki/Multi-tenant
[7] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud?denominazione=saas
[8] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud?denominazione=paas
[9] https://it.wikipedia.org/wiki/Application_programming_interface
[10] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud?denominazione=iaas

- **cloud ibrido:** combinazione del modello pubblico e di quello privato, ovvero un modello in cui l'utente utilizza risorse sia del suo cloud privato che di un cloud pubblico. Ad esempio, un utente che dispone di un cloud privato, può utilizzare le risorse di un cloud pubblico per gestire improvvisi picchi di lavoro che non possono essere soddisfatti facendo ricorso unicamente alle risorse disponibili nel cloud privato. Tuttavia questo modello comporta comunque la proprietà e conseguente gestione della parte privata delle infrastrutture, risultando in maggiori costi e rischi. Ulteriori informazioni nella definizione di cloud ibrido[11].

Le linee guida in questo documento saranno orientate al modello di cloud pubblico, in quanto più allineato agli obiettivi della strategia di cloud enablement, ovvero:

- minimizzazione dell'infrastruttura proprietaria di un ente
- riduzione dell'onere di manutenzione ed ammodernamento continuo dell'infrastruttura e delle competenze necessarie per farlo
- incremento della facilità di adozione delle soluzioni e dei servizi progressivamente disponibili grazie all'evoluzione di mercato
- adozione di servizi ai più alti standard qualitativi di mercato, riflessi negli SLA garantiti

È possibile consultare le definizioni del modello cloud e le proprietà specifiche dei servizi facendo riferimento al documento del NIST[12].

Non tutti i servizi e le infrastrutture di cloud computing sono uguali. In alcuni casi tali servizi possono anche non rispettare i principali standard di sicurezza, garanzie operative e affidabilità definiti a livello internazionale. Questa disomogeneità può rappresentare un rischio quando si affidano i propri dati a provider che non garantiscono dei livelli minimi di sicurezza e affidabilità.

Il modello Cloud della PA[13] consente di mitigare tale rischio, qualificando servizi e infrastrutture cloud secondo specifici parametri di sicurezza e affidabilità idonei per le esigenze della pubbliche amministrazioni. Il Cloud della PA si compone di infrastrutture e servizi, qualificati da ACN sulla base di un insieme minimo di requisiti, che possono essere confrontati e consultati sul Cloud Marketplace[14].

Il sito cloud.italia.it[15] contiene un approfondimento sulla qualificazione dei servizi cloud[16] e sul catalogo dei servizi cloud qualificati[17].

2.2 1.2 Perché usare il cloud

Il cloud rappresenta un grande cambiamento rispetto alla visione tradizionale delle pubbliche amministrazioni in materia di risorse IT. Per una vasta gamma di servizi e sistemi, che vanno dalla sicurezza informatica alla produttività individuale e all'archiviazione, le soluzioni cloud rappresentano spesso la soluzione più vantaggiosa per diversi motivi.

2.2.1 1.2.1 Riduzione dei costi

Le applicazioni che utilizzano risorse hardware on-premise richiedono un investimento iniziale significativo, anche se il software utilizzato è gratuito o open source. Oltre ad essere necessari per ospitare anche il software gestionale più semplice, data center, reti, server, storage e sistemi operativi richiedono investimenti, tempo e personale dedicato per garantirne il corretto funzionamento e il continuo aggiornamento nel rispetto di standard qualitativi e di sicurezza.

[11] https://cloud.italia.it/glossario/#cloud-ibrido
[12] https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
[13] https://docs.italia.it/italia/piano-triennale-ict/cloud-docs/it/stabile/index.html
[14] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud
[15] https://cloud.italia.it
[16] https://cloud.italia.it/qualificazione-servizi-cloud/
[17] https://cloud.italia.it/catalogo-servizi-cloud/

Il cloud elimina le spese in conto capitale necessarie per l'acquisto di hardware e i costi legati alla gestione e manutenzione dei data center di proprietà delle amministrazioni. Le applicazioni in cloud, infatti, richiedono investimenti iniziali estremamente limitati e si pagano generalmente in base al consumo, consentendo così di gestire la crescita di un servizio in maniera dinamica. La decisione di migrare verso una nuova soluzione non è, quindi, condizionata da eventuali investimenti già fatti. Allo stesso tempo, poiché si paga solo il consumo della risorsa, quando un servizio non è più utilizzato, non è più un costo.

Inoltre, le applicazioni basate su hardware on-premise richiedono un piano di investimenti che deve tener conto dei prezzi riferiti al momento della sottoscrizione del contratto e di alcuni anni di manutenzione e supporto. I costi complessivi, per es. licenze, energia elettrica, potenza di calcolo, manodopera e così via, raramente diminuiscono nel corso della durata del servizio. Al contrario, i servizi cloud tendono ad essere sempre più economici per le dinamiche di mercato, infatti, la pressione competitiva, l'hardware ottimizzato e l'aumento dei tassi di utilizzo stanno riducendo progressivamente i costi delle applicazioni in cloud e delle infrastrutture virtuali.

Infine, i provider di servizi cloud offrono generalmente strumenti di monitoraggio e alerting che facilitano un controllo continuo e una gestione efficiente dei costi. È possibile, ad esempio, monitorare più precisamente i costi associati ai singoli servizi e l'eventuale presenza di "risorse zombie", cioè risorse infrastrutturali non usate e che possono essere spente, risparmiando sul costo associato. Per un approfondimento su questo tema si veda il capitolo 5.3.8.

In generale, a seconda della categoria di cloud computing scelta (vedi sopra: SaaS, PaaS, IaaS), del tipo di applicativo da migrare, della rispettiva strategia di migrazione e del contesto in cui l'amministrazione opera, la riduzione dei costi sarà più o meno consistente. Si consiglia di far riferimento al capitolo 3.3 per pianificare l'ottimizzazione dei costi prima di una migrazione identificando le aree ed i componenti su cui è possibile ottenere un vantaggio in termini di costo una volta in cloud rispetto alla situazione corrente.

2.2.2 1.2.2 Elasticità reale

Le soluzioni IT on-premise, anche quando sono scalabili, hanno dei limiti. Ad esempio, è necessario pianificare investimenti e sforzi costanti per mantenere i margini sufficienti di scalabilità ed evitare situazioni di sottodimensionamento, causando disservizi, o sovradimensionamento, causando uno spreco di risorse e costi maggiori. Per poter garantire la vera elasticità, è necessario mantenere costantemente attivo un grande surplus di risorse che rimangono tuttavia non utilizzate per la maggior parte del tempo.

A differenza delle soluzioni on-premise, i servizi cloud sono davvero elastici: le risorse di calcolo, storage o rete possono essere consumate solo quando richiesto e dismesse quando non sono più necessarie, riducendo la complessità nella pianificazione della capacità dell'infrastruttura IT.

Ridimensionare le risorse in modo realmente elastico significa, infatti, fornire la giusta quantità di risorse IT (ad esempio maggiore o minore potenza di calcolo, risorse di archiviazione e larghezza di banda) proprio quando è necessario.

Infine, il paradigma cloud non richiede alcun investimento a lungo termine sull'hardware e non comporta quello spreco di risorse determinato dalla sottoutilizzazione della capacità.

2.2.3 1.2.3 Facilità degli aggiornamenti

Le soluzioni IT commerciali o auto-sviluppate in locale richiedono finanziamenti, risorse umane, impegno e pianificazione per poter essere aggiornate costantemente. Il supporto e gli aggiornamenti sono attività costose e complicate da gestire ed è molto difficile per qualsiasi organizzazione tenere il passo con la costante richiesta di aggiornamenti e patch di sicurezza. Ne consegue che, spesso, le infrastrutture della pubblica amministrazione non vengono adeguatamente aggiornate.

In cloud invece, a seconda della categoria di cloud computing utilizzata (vedi sopra: SaaS, PaaS, IaaS), la responsabilità di queste dinamiche viene ridistribuita tra fornitore e amministrazione.

Nello specifico in caso di servizi SaaS, l'aggiornamento del software e dell'infrastruttura è completamente demandata al fornitore ed incluso nei costi.

Per servizi PaaS e IaaS invece, la responsabilità dell'amministrazione ricade nell'ambito della gestione dell'applicativo e degli accessi alle piattaforme, mentre i fornitori si occupano di garantire l'aggiornamento delle risorse infrastrutturali e degli ambienti.

2.2.4 1.2.4 Riduzione complessità del supporto

I servizi IT tradizionali spesso dipendono dal software installato sul computer dell'utente. Quest'ultimo deve essere gestito insieme a tutte le altre applicazioni dell'utente. In molti casi, si rende necessario soddisfare dipendenze applicative molto specifiche legate alle versioni del sistema operativo e degli aggiornamenti di sistema affinché il software funzioni correttamente.

Gli aggiornamenti devono essere testati prima di essere applicati su un numero elevato di sistemi e, a volte, un'applicazione obsoleta può rallentare l'adozione di nuovi sistemi operativi e di applicazioni più moderne.

I servizi cloud sono progettati per essere fruibili tramite internet. Per rimanere sul mercato, i fornitori devono aggiornare i propri servizi per supportare le ultime versioni dei browser, i sistemi operativi e le scelte dei dispositivi dei propri utenti.

Per una pubblica amministrazione che gestisce applicazioni che devono essere fruibili tramite numerose tipologie di dispositivi, come laptop, desktop e dispositivi mobili, una qualsiasi soluzione che riduca la quantità di lavoro necessaria a mantenere il software aggiornato rappresenta un gran vantaggio.

2.2.5 1.2.5 Riduzione delle attività manuali a basso valore aggiunto

I data center on-premise richiedono in genere un impegno notevole nell'organizzazione e nell'assemblaggio dei rack[18], nella configurazione degli apparati (ad esempio: server, storage, apparati di networking), nell'applicazione di patch software e altre attività di gestione IT dispendiose in termini di tempo.

La semplicità con cui si possono aumentare o ridurre le risorse necessarie in cloud ad esempio, permetterà alle amministrazioni di alleggerire notevolmente tutte quelle operazioni a basso valore aggiunto e il lungo iter burocratico necessario per il provisioning di risorse aggiuntive.

Grazie alla facilità degli aggiornamenti e del supporto semplificato dei servizi cloud, invece, l'amministrazione non ha bisogno di aggiornare i sistemi operativi dei server, acquistare hardware, contrattualizzare personale esterno, pianificare le operazioni o migrare i dati per ottenere i benefici della tecnologia più recente.

Infine, il cloud facilita l'adozione di pratiche di automazione che aumentano la ripetibilità di operazioni critiche e permettono di accelerare i processi di delivery, riducendo al minimo la possibilità di errori o configurazioni errate aumentando il controllo sui processi. Per un approfondimento su questo tema si rimanda al capitolo 5.3.9.

2.2.6 1.2.6 Adeguamento normativo in termini di sicurezza e privacy

Amministrare le infrastrutture IT comporta responsabilità non solo di tipo economico-amministrativo, ma soprattutto di sicurezza e di protezione dei dati personali. Le recenti normative in materia di privacy e di sicurezza informatica (ad es. GDPR[19] e misure minime di sicurezza informatica ICT per la pubblica amministrazione[20]) impongono infatti anche alle pubbliche amministrazioni l'adozione di misure tecniche e organizzative adeguate a garantire la sicurezza del trattamento dei dati.

[18] https://it.wikipedia.org/wiki/Rack_(informatica)
[19] https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_it
[20] https://www.agid.gov.it/it/sicurezza/misure-minime-sicurezza-ict

Molti cloud service provider (CSP) offrono un'ampia gamma di metodi, tecnologie e controlli che rafforzano la sicurezza complessiva, grazie alla protezione dei dati (che possono essere criptati con i più alti livelli di sicurezza del mercato), dell'applicazione e dell' infrastruttura da minacce potenziali. I CSP qualificati da ACN e consultabili sul Cloud Marketplace[21] hanno infrastrutture e servizi sviluppati secondo criteri di affidabilità e sicurezza considerati necessari per i servizi digitali della PA. Ad esempio, i data center dei CSP hanno tutti la certificazione ISO/IEC 27001[22]. Questo e altri criteri di qualificazione in termini di sicurezza dei CSP possono essere consultati nel Kit Percorsi di qualificazione CSP e SaaS[23].

I CSP offrono anche servizi specifici di disaster recovery (vedi capitolo 5.3.10), che permettono di ripristinare più rapidamente i sistemi IT maggiormente critici senza sostenere le spese di un secondo sito fisico. Questo migliora la continuità operativa dell'infrastruttura e mitiga il rischio di perdita di dati. Inoltre, le applicazioni cloud mettono solitamente a disposizione strumenti di auditing e controllo delle informazioni che consentono di intervenire puntualmente in caso di problemi.

Certamente la responsabilità finale della sicurezza e dell'integrità dei dati è sempre dell'amministrazione: gli strumenti cloud possono semplificare la stesura e la messa in campo di piani di sicurezza e continuità operativa ma non li sostituiscono.

Non basta dotarsi di soluzioni cloud per assicurare privacy ai propri utenti e sicurezza delle infrastrutture e servizi IT, bensì serve un processo continuo di vigilanza e controllo che agisca trasversalmente su tutte le aree di interesse fin dalla prima fase di progettazione dei servizi; e che sia costantemente aggiornato rispetto allo stato dell'arte delle principali misure di sicurezza.

2.2.7 1.2.7 Miglioramento dei servizi

Sfruttando le potenzialità del cloud, le pubbliche amministrazioni hanno l'opportunità di migliorare la qualità dei propri servizi, siano questi ad uso interno o ad uso del cittadino.

Grazie al cloud, l'amministrazione può gestire i servizi in maniera più efficiente ed efficace, riuscendo a concentrarsi maggiormente sulle funzionalità da offrire ai propri utenti.

Prima di tutto, il cloud garantisce un rapido accesso a tutte le informazioni indipendentemente dalla propria postazione fisica. I dati sono infatti accessibili ovunque, attraverso una molteplicità di device e secondo standard di sicurezza elevati, presentato precedentemente.

Inoltre, l'adozione del cloud favorisce l'uso di architetture moderne, basate su principi tecnologici avanzati come ad esempio il basso accoppiamento dei componenti e il "design for failure" (per approfondimento si veda il capitolo 5.4.3), lontani dalla struttura monolitica degli applicativi legacy. Questo rende gli applicativi molto più adeguati alle necessità di interoperabilità e comunicazione tra diversi servizi (e tra le rispettive basi di dati). Le soluzioni SaaS dei cloud service provider (CSP) qualificati da ACN e consultabili sul Cloud Marketplace[24], ad esempio, offrono tutte uno strato di interoperabilità fruibile tramite API. Questo permette di avere maggiore flessibilità nel provare nuovi servizi o apportare modifiche.

Nel caso un applicativo debba essere scomposto nelle sue parti prima di essere migrato al cloud (si veda il capitolo 4.1.6 per la strategia di migrazione Re-architect), l'amministrazione ha la possibilità di usare questo cambiamento come occasione per ridisegnare il servizio anche nel suo processo per renderlo più adatto alle esigenze degli utenti (e per questa finalità si consiglia di consultare le linee guida che si trovano sul sito di designers.italia.it[25]).

Infine, grazie alla scalabilità reale del cloud, si ha un miglioramento dell'accessibilità e della disponibilità dei servizi. Usando applicativi in cloud, l'amministrazione può assicurarsi che tali applicativi siano disponibili anche durante i picchi di accesso. Ad esempio, nel caso di un servizio con picchi di traffico solo in determinati periodi dell'anno

[21] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud
[22] https://it.wikipedia.org/wiki/ISO/IEC_27001
[23] https://www.agid.gov.it/sites/default/files/repository_files/kit_percorsi_di_qualificazione_id2.pdf
[24] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud
[25] https://designers.italia.it/guide/

(come il servizio di iscrizione alle scuole elementari dove i genitori accedono e iscrivono i propri figli solo ad inizio anno) si ha la sicurezza di non incorrere in downtime o momenti di disservizio durante i periodi di maggiore carico.

CAPITOLO 3

2. Come iniziare

Iniziare è spesso lo scoglio più grande da superare quando si decide di intraprendere una nuova attività o di adottare un nuovo paradigma come quello cloud. Proprio per questo, l'obiettivo primario è di fornire delle linee guida che aiutino le pubbliche amministrazioni a fare il primo passo verso il cloud, e a pianificare efficacemente la migrazione dei loro servizi.

In questo capitolo, oltre ad informazioni molto operative sui contatti ed il procurement, offriamo una visione d'insieme sull'approccio da attuare per iniziare a migrare al cloud l'insieme degli applicativi (e i rispettivi servizi) che la pubblica amministrazione gestisce. Ogni fase dell'approccio verrà poi approfondita nei capitoli successivi del documento.

3.1 2.1 Roadmap di una migrazione

La migrazione dell'intero parco applicativo al cloud è un'operazione complessa che riguarda aspetti tecnologici, di processo e culturali.

È cruciale per il successo dell'operazione iniziare a beneficiare della nuova architettura durante il percorso, gradualmente, e non solo al termine dell'intera transizione.

Per raggiungere questo risultato e contestualmente ridurre i rischi legati a questa sfida, è fondamentale procedere in modo iterativo ed incrementale partendo dagli applicativi che traggono un beneficio significativo dall'adozione del paradigma cloud, che al contempo rappresentano un rischio ridotto per la continua erogazione dei servizi supportati e che risultano relativamente semplici da migrare.

Questo approccio permette al team di lavoro di scoprire ed affrontare le problematiche che emergono strada facendo, senza particolari pressioni legate alla criticità dell'applicativo. La conoscenza che si sviluppa nel superare queste sfide è poi di supporto quando si devono affrontare applicativi a rischio maggiore: la conoscenza acquisita con le migrazioni precedenti infatti riduce il rischio delle migrazioni successive.

Le migrazioni iniziali devono anche contribuire a creare l'evidenza del valore del cloud e la fiducia necessaria a procedere con le successive migrazioni includendo i cambiamenti che possono essere richiesti a livello di processi, attività o responsabilità.

Riassumendo la strategia di migrazione da adottare in una roadmap, ovvero in un percorso che permetta di definire in modo chiaro gli obiettivi di ogni fase, possiamo identificare tre momenti rilevanti:

1. **Ora**: ovvero la fase iniziale focalizzata sulla creazione dei primi casi di successo con applicativi scelti secondo specifici criteri di prioritizzazione (vedi capitolo 3.1.2)
2. **Subito dopo**: ovvero una seconda fase con obiettivi da conseguire a seguito dell'esperienza fatta nella prima fase e degli apprendimenti e della conoscenza maturata su: gli aspetti specifici della piattaforma cloud selezionata come destinazione, i vincoli incontrati e le problematiche specifiche emerse durante la migrazione degli applicativi rispetto al contesto di partenza
3. **Più tardi**: ovvero un'ultima fase in cui si va a concludere il processo forti dell'esperienza e dei successi conseguiti nelle fasi precedenti

Le tre fasi identificate suggeriscono un approccio multi-fase che può poi essere adattato alle specifiche realtà.

OBIETTIVO: Creare i **primi casi di migrazione di successo** mostrando il valore che si ottiene dalla nuova infrastruttura	OBIETTIVO: Sfruttando le conoscenze maturate nella fase precedente, creare altri **casi di successo con migrazioni che mostrano l'alto valore dalla migrazione**, ma **più impegnative** dal punto di vista del **rischio** o della **complessità di esecuzione**	OBIETTIVO: Concludere la **migrazione degli applicativi rimanenti**, **più rischiosi e più complicati** forti delle esperienze precedenti
COME: 1. Identificare gli applicativi che possono ottenere maggiore beneficio dall'adozione del cloud con un rischio ridotto per quanto riguarda la criticità dei servizi che erogano e la relativa semplicità di migrazione 2. Identificare le strategie di migrazione applicabili 3. Valutare le competenze necessarie per attuare le strategie identificate 4. Effettuare la migrazione al miglior rapporto costi/benefici e validarne il risultato	COME: 1. Identificare gli applicativi che possono cogliere beneficio dall'adozione del cloud con un rischio medio o una semplicità di migrazione media 2. Identificare le strategie di migrazione applicabili 3. Valutare le competenze necessarie per attuare le strategie identificate 4. Effettuare la migrazione al miglior rapporto costi/benefici e validarne il risultato	COME: 1. Per gli applicativi restanti, identificare le strategie di migrazione applicabili 2. Identificare le strategie di migrazione applicabili 3. Valutare le competenze necessarie per attuare le strategie identificate 4. Effettuare la migrazione al miglior rapporto costi/benefici e validarne il risultato

Per iniziare questo percorso, ovvero per identificare gli applicativi da cui iniziare, pianificarne ed eseguirne la migrazione, suggeriamo di seguire un approccio articolato in più step (da ripetersi poi in maniera iterativa e incrementale come illustrato sopra):

1. **Lista degli applicativi e classificazione dei servizi attivi**: un primo passo che consiste nello stilare una lista degli applicativi attualmente in uso, ovvero sia gli applicativi utilizzati abitualmente che quelli con accessi saltuari o legati a specifiche necessità. L'obiettivo è di avere una visione d'insieme degli applicativi e dei

rispettivi servizi che l'amministrazione gestisce. Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale e i responsabili IT dell'amministrazione. Questa prima attività è presentata in dettaglio nel capitolo 3.1.1. Una volta definito l'elenco dei servizi è necessario procedere con la classificazione di dati e servizi seguendo la procedura descritta su PA digitale 2026[26].

2. **Prioritizzazione degli applicativi**: identificare gli applicativi candidati ad essere migrati nell'immediato classificandoli secondo quattro livelli che aiutano una valutazione orientata al valore generato, bilanciato rispetto al rischio potenziale ed alla difficoltà dell'operazione. L'obiettivo è di razionalizzare il panorama degli applicativi e identificare quelli prioritari da cui partire con la migrazione al cloud (se confermato dalle fasi successive a questa). Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale, il centro di competenza, l'unità di esecuzione, i responsabili IT dell'amministrazione, interloquendo con i responsabili dei servizi per la valutazione delle opportunità e dei rischi. Il framework di prioritizzazione è illustrato nel capitolo 3.1.2

3. **Scheda di assessment dell'applicativo**: approfondire gli aspetti e le caratteristiche tecnologiche e non degli applicativi identificati come prioritari attraverso la compilazione di una scheda di assessment. L'obiettivo è di raccogliere ad un sufficiente livello di dettaglio le informazioni necessarie a supportare un processo decisionale informato sulle possibili strategie da applicare, come descritto successivamente. Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale, il centro di competenza, l'unità di esecuzione e i responsabili IT dell'amministrazione interloquendo con i responsabili dei servizi per la valutazione dei bisogni dell'applicativo in analisi. A questa parte è dedicato il capitolo 3.2

4. **Identificazione delle strategie di migrazione possibili**: identificare quali strategie di migrazione, tra le sei possibili, siano più adatte per ciascun applicativo sulla base della scheda di assessment. L'obiettivo è di evidenziare le diverse opzioni disponibili prima di procedere con la scelta di quale adottare. Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale, il centro di competenza, l'unità di esecuzione e i responsabili IT dell'amministrazione ed eventuali fornitori. Le strategie di migrazione sono trattate nel capitolo 4.1. La scelta del modello di cloud deployment (pubblico, pubblico criptato, private/hybrid su licenza, private) deve essere effettuata in coerenza con l'esito della classificazione di dati e servizi e di quanto indicato nel documento Strategia Cloud Italia[27].

5. **Analisi costi-benefici**: per ciascuna delle strategie di migrazione identificate come possibili per l'applicativo effettuare un'analisi costi-benefici per valutarne l'opportunità. L'obiettivo è identificare il modello cloud migliore in base al contesto e alle circostanze in cui l'amministrazione si trova. Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale, il centro di competenza, l'unità di esecuzione e i responsabili IT dell'amministrazione. Come svolgere un'analisi costi-benefici è spiegato nel capitolo 3.3

6. **Valutazione delle competenze**: uno dei fattori cruciali per il successo di un processo di migrazione sono le competenze necessarie. Attraverso uno strumento di assessment delle competenze stimoliamo la riflessione sulle competenze necessarie rispetto a quelle disponibili, coprendo non solo l'ambito tecnologico ma tutti quelli che possono essere necessari per il successo del processo di migrazione. Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale, il centro di competenza, l'unità di esecuzione e i responsabili IT dell'amministrazione. Per evitare il rischio lock-in, l'amministrazione deve prendersi carico delle responsabilità e delle competenze rispetto sia al centro di competenza che ad eventuali fornitori. Questo step comprensivo di pianificazione delle competenze e degli aspetti ad esse connessi è trattato nelle sezioni 4.2 e 4.4

7. **Scelta della strategia e pianificazione della migrazione**: sulla base delle considerazioni fatte con l'analisi costi-benefici e la valutazione delle competenze scegliere quale strategia di migrazione effettivamente usare. L'obiettivo è di prendere una decisione informata e pianificare in maniera adeguata la migrazione. Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale, i centri di competenza, l'unità di esecuzione, i responsabili IT dell'amministrazione ed eventuali fornitori. Le strategie di migrazione e gli altri aspetti da prendere in considerazione una volta scelta la strategia di migrazione (ad es. SLA richiesti ai fornitori, come evitare il rischio lock-in) sono trattati nel capitolo 4

[26] https://padigitale2026.gov.it/come-partecipare/classifica-pa/

[27] https://docs.italia.it/italia/cloud-italia/strategia-cloud-italia-docs/it/stabile/4_la_strategia_cloud_per_la_pubblica_amministrazione.html#la-classificazione-dei-dati-e-dei-servizi

8. **Esecuzione della migrazione**: ovvero il passo cruciale durante il quale si esegue l'effettiva migrazione dell'applicativo a più alta priorità. In questa fase sarà fondamentale il supporto del centro di competenza, in quanto aggregatore di conoscenza quindi in grado sia di ricoprire un ruolo di advisor per l'amministrazione durante il processo che di consolidare la conoscenza che l'amministrazione acquisisce per condividerla poi con l'unità di controllo. Si consiglia pertanto di coinvolgerlo continuamente durante l'esecuzione della migrazione, insieme al responsabile per la trasformazione digitale, ai responsabili IT e ai fornitori. All'esecuzione della migrazione sono dedicati due interi capitoli, il 5 e il 6
9. **Check dei risultati**: l'ultimo step riguarda la riflessione sui risultati raggiunti e sull'impatto generato dall'operazione di migrazione. L'obiettivo è di valutare i progressi fatti e il valore ottenuto migrando al cloud anche calcolando e interpretando alcuni indicatori di risultato. Si consiglia di svolgere questa attività con il responsabile per la trasformazione digitale, il centro di competenza e i responsabili IT dell'amministrazione. Gli indicatori di risultato post-migrazione sono approfonditi nel capitolo 7

Una visione di alto livello dell'approccio con i macro-obiettivi e i rispettivi step (attività) è rappresentata nella figura sotto.

Una visione in dettaglio dell'approccio è invece rappresentata nelle immagini seguenti, dove per ogni macro-obiettivo e per i rispettivi step abbiamo evidenziato anche le persone da coinvolgere e il risultato (output) atteso.

Primo macro-obiettivo:

Secondo macro-obiettivo:

Terzo macro-obiettivo:

Quarto macro-obiettivo:

Quinto macro-obiettivo:

In generale, migrare al cloud richiede un esercizio di gestione e orchestrazione del cambiamento che va oltre la semplice e diligente applicazione di strumenti e metodologie. Di questo bisogna essere coscienti ancor prima di iniziare. La complessità di questo processo di trasformazione è insita nella natura della sfida stessa, costituita da un insieme di fattori (tecnologia, persone, contesto, pratiche, ecc.) connessi tra loro e non separabili nè attaccabili separatamente. Una sfida complessa non può essere affrontata con un approccio analitico. Essa ha piuttosto bisogno

MACRO OBIETTIVO	INDIVIDUARE **APPLICATIVI** CHE POSSONO COGLIERE **MAGGIORE BENEFICIO DAL CLOUD** CON UN **RISCHIO RIDOTTO**	
ATTIVITÀ	**1. LISTA APPLICATIVI**	**2. PRIORITIZZAZIONE**
PERSONE DA COINVOLGERE	○ Responsabili IT dell'amministrazione ○ RTD	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza ○ Unità di esecuzione
OUTPUT	○ Elenco degli applicativi con le informazioni richieste dall'allegato Lista degli applicativi (capitolo 3.1)	○ Elenco applicativi ordinato per priorità secondo opportunità da cogliere, basso rischio,facile da migrare o altro (capitolo 3.1)

MACRO OBIETTIVO	INDIVIDUARE LE **STRATEGIE DI MIGRAZIONE POSSIBILI** PER L'**APPLICATIVO PRIORITIZZATO**	
ATTIVITÀ	**3. SCHEDA ASSESSMENT**	**4. STRATEGIE POSSIBILI**
PERSONE DA COINVOLGERE	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza ○ Unità di esecuzione
OUTPUT	○ Scheda di assessment compilata per ogni applicativo prioritizzato (capitolo 3.2)	○ Identificazione strategie di migrazione possibili per l'applicativo come da istruzioni nel capitolo 4.1

MACRO OBIETTIVO	VALUTARE **OPPORTUNITÀ** E **FATTIBILITÀ** DELLA MIGRAZIONE DELL'APPLICATIVO E DECIDERE **COME MIGRARLO**		
ATTIVITÀ	**5. COSTI-BENEFICI**	**6. COMPETENZE**	**7. SCELTA & PIANIFICAZIONE MIGRAZIONE**
PERSONE DA COINVOLGERE	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza ○ Unità di esecuzione ○ Eventuali fornitori
OUTPUT	○ Analisi costi-benefici per ogni strategia possibile come descritta nel capitolo 3.3	○ Elenco dei gap identificati dalla valutazione competenze (capitolo 4.2)	○ Migrazione da applicare e pianificazione come da istruzioni nel capitolo 4

MACRO OBIETTIVO	**ESEGUIRE LA MIGRAZIONE** DELL'APPLICATIVO SCELTO **COLLABORANDO** CON IL **CENTRO DI COMPETENZA**	
ATTIVITÀ	**8. ESECUZIONE MIGRAZIONE**	
PERSONE DA COINVOLGERE	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza ○ Unità di esecuzione ○ Eventuali fornitori	
OUTPUT	○ Validazione della migrazione (capitoli 5 e 6) e condivisione dell'esperienza con il centro competenze (capitolo 7.4)	

MACRO OBIETTIVO	RIFLETTERE SULLA **PERFORMANCE** E I **RISULTATI** RAGGIUNTI **DOPO LA MIGRAZIONE**
ATTIVITÀ	**9. CHECK RISULTATI**
PERSONE DA COINVOLGERE	○ Responsabili IT dell'amministrazione ○ RTD ○ Centro di competenza
OUTPUT	○ Elenco delle azioni da intraprendere a seguito del controllo degli indicatori nel capitolo 7.1

di un approccio emergente e di una buona governance che affronti il problema nella sua interezza, considerando tutti i fattori coinvolti e osservando l'evoluzione nel tempo della relazione tra di essi a seconda della soluzione applicata. Di conseguenza, oltre a iniziare il viaggio seguendo un percorso predefinito per il fattore tecnologia, le pubbliche amministrazioni dovranno al contempo impegnarsi in attività che impattano sugli altri fattori correlati, ovvero sui processi, sulle persone e in generale sulla cultura dell'organizzazione.

Per esempio, nel Comune di Milano, dove questo percorso verso il cloud è già stato iniziato, insieme ad una roadmap per migrare diversi servizi è stato creato e promosso dalla Direzione Sistemi Informativi e Agenda Digitale un programma di comunicazione per il cambiamento chiamato "Innesco" per condividere a tutti i livelli nuovi modelli e strumenti per ripensare la system integration e lo sviluppo applicativo secondo i paradigmi moderni come quello cloud.

Sulla stessa linea, in Corte dei Conti, è stata introdotta una strategia di comunicazione mirata a raccontare il valore dei nuovi applicativi su cloud in maniera semplice e divertente tramite video e animazioni inviati a tutti i dipendenti con una newsletter settimanale.

CAPITOLO 4

3. Assessment servizi e infrastruttura

In questo capitolo illustriamo nel dettaglio i primi tre step del percorso di abilitazione al cloud presentato nel capitolo 2.

Il focus iniziale è sulla mappatura degli applicativi e dei servizi attivi e sulla loro prioritizzazione ed a seguire, sulla creazione di una scheda di assessment di dettaglio per ciascun applicativo prioritizzato. Infine, vengono fornite indicazioni su come svolgere un'analisi costi-benefici in dettaglio e sugli aspetti da considerare in termini di gestione delle licenze.

Per svolgere questi esercizi forniamo strumenti specifici, a cui facciamo riferimento nei rispettivi capitoli e che possono essere consultati in allegato a questo documento. Come anticipato nel capitolo 2.3, si ricorda che l'esecuzione di queste attività deve essere coadiuvata da opportune competenze (provenienti dal centro di competenza, se esistente, o dall'unità di esecuzione) se non presenti all'interno dell'amministrazione

L'obiettivo è di aiutare la pubblica amministrazione ad identificare quegli applicativi migrabili con maggiori benefici e minori criticità e che dunque rappresentano un'opportunità da cogliere per iniziare con la migrazione al cloud.

4.1 3.1 Costruire una mappa degli applicativi e dei servizi attivi

Il primo obiettivo da raggiungere è quello di avere una visione d'insieme degli applicativi e di classificarli in modo semplice per poterli prioritizzare ed identificare quelli da cui iniziare, o con cui procedere, la migrazione al cloud, ovvero quelli che ne traggono maggiori benefici e pongono minori criticità.

4.1.1 3.1.1 Lista degli applicativi

Il primo passo da compiere è stilare una lista degli applicativi attualmente in uso (vedi allegato «Lista degli applicativi in formato ODS» o «Lista degli applicativi in formato XLXS»), ovvero sia gli applicativi utilizzati abitualmente che quelli con accessi saltuari o legati a specifiche necessità. Sono da considerare tutti gli applicativi che utilizzano l'infrastruttura dell'organizzazione.

Per ogni applicativo è utile tracciare il nome con la versione, l'interlocutore con cui si interagisce per gestire gli aggiornamenti, le evoluzioni, ecc. (può essere il produttore stesso dell'applicativo, il fornitore, includendo altre entità pubbliche come un ministero) e le eventuali licenze associate (vedi il capitolo 3.1.2).

Ad ogni applicativo è poi utile associare i servizi che supporta e l'indicazione delle persone di riferimento per tali servizi. Con applicativo, infatti, si fa riferimento al software o alle componenti strettamente tecnologiche dell'applicazione. Con servizio, invece, si intende la prestazione erogata agli utenti, siano questi esterni (ad es.i cittadini) o interni (ad es. i dipendenti) all'amministrazione. Un applicativo può dunque supportare uno o più servizi e, viceversa, un servizio può richiedere uno o più applicativi.

Si consiglia perciò di prestare particolare attenzione alla mappatura dei servizi sulla lista degli applicativi, in quanto l'amministrazione dovrà probabilmente considerare di migrare più applicativi per iniziare ad erogare un servizio con il paradigma cloud.

La costruzione di questo elenco può seguire un approccio iterativo ed incrementale, in cui si collezionano inizialmente gli applicativi di cui si ha maggiore evidenza e si integra successivamente la lista con quelli per i quali è necessaria un'indagine più approfondita per verificarne l'esistenza e l'utilizzo.

4.1.2 3.1.2 Licenze

3.1.2.1 Licenze software

Nella compilazione della lista degli applicativi, risulta utile riportare anche le informazioni relative alle licenze a loro associate, in quanto saranno uno dei fattori da considerare nello step di prioritizzazione. È comunque importante fare attenzione a non considerare le licenze software come unico driver di prioritizzazione per la migrazione in cloud (vedi capitolo 3.1.3 per i dettagli).

Le informazioni relative alle licenze da inserire durante la compilazione della lista degli applicativi sono:

- tipologia di licenza on-premise e open source
- data di scadenza contrattuale della licenza

3.1.2.1.1 Tipologie di licenze on-premise

Vi sono diverse tipologie di licenze on-premise, quali ad esempio:

- **licenza perpetua**: in passato l'unica opzione disponibile era l'acquisto di software su supporto fisico come un floppy o disco ottico. Avere un programma su supporto fisico rendeva facile installarlo su eventuali nuove macchine. Software di questo tipo erano distribuiti tramite una licenza perpetua garantendo al titolare la possibilità di usarlo per tutto il tempo che desidera senza costi aggiuntivi in base ai termini di contratto di licenza con l'utente finale o EULA. Questo tipo di licenza è generalmente meno comune oggi, in quanto le software house sviluppano i loro programmi con una strategia che favorisce un'alta frequenza di aggiornamento solitamente compresa nel prezzo.
- **licenza per sito**: un tipo di licenza software che consente all'utente di installare un pacchetto software in più computer contemporaneamente, ad esempio in un particolare sito (struttura) o in un ente. A seconda dell'importo delle tariffe pagate, la licenza può essere illimitata o può limitare l'accesso simultaneo a un determinato numero di utenti. Il termine «sito» si intende a definire una limitazione sui diritti di accesso dell'utente. Al giorno d'oggi, questi tipi di licenze sono rare, ma ancora utilizzate in alcuni settori. I venditori possono inserire clausole che consentono ai rappresentanti di visitare il sito e verificare che l'uso del software confermi la licenza (auditing). Una licenza per sito funziona in maniera molto simile a una licenza perpetua tranne che si applica a tutti i membri del "sito". Solitamente il prezzo per utente diminuisce con l'aumentare degli utenti.
- **licenza nominativa**: una licenza per postazione è un modello di licenza software basato sul numero di singoli utenti che hanno accesso a un servizio o prodotto digitale. Ad esempio, la licenza per postazione da 50 utenti permette fino a 50 utenti nominati individualmente di accedere al programma. Un'alternativa è la licenza utente simultanea, basata sul numero di utenti simultanei, indipendentemente da quali individui stiano accedendo al programma. Ad esempio, in una licenza di utilizzo simultaneo da 50 utenti, dopo che 50 utenti sono connessi

al programma, il 51° utente viene bloccato. Quando uno dei primi 50 si disconnette, l'utente successivo può accedere.

3.1.2.1.2 Tipologie di licenze open source

Vi sono diverse tipologie di licenze open source e solitamente i software licenziati in questo modo non hanno particolari restrizioni quando migrati in cloud.

Per un compendio più completo sui tipi di licenze e sulla loro gestione per il riuso degli applicativi, rimandiamo al documento Allegato C: Guida alle licenze Open Source[28] pubblicato da AgID.

4.1.3 3.1.3 Prioritizzazione degli applicativi

Come decidere ora l'ordine con cui migrare le applicazioni nel cloud? È una domanda molto importante, perché un successo iniziale durante la migrazione al cloud è fondamentale per continuare il percorso di adozione del cloud. Viceversa, un insuccesso precoce può pregiudicare la prosecuzione.

Benché i vantaggi del cloud siano chiari, osservare e analizzare in dettaglio gli aspetti di ogni applicazione che è stata creata o implementata nell'organizzazione può essere complicato e dispendioso in termini di tempo. Sebbene non esista una risposta valida per tutti gli scenari, esistono alcune buone pratiche che è possibile utilizzare per iniziare a fare una valutazione di alto livello sull'ordine con cui migrare gli applicativi. Questo tipo di pianificazione anticipata può semplificare il processo di migrazione e rendere più fluida l'intera transizione cloud.

Il seguente framework aiuta ad identificare l'ordine con cui procedere con la valutazione di dettaglio (vedo 3.2) per la migrazione degli applicativi. Esso si basa su quattro livelli di priorità come illustrato nel seguente grafico:

I software con "opportunità da cogliere" hanno una priorità maggiore rispetto a quelli con "rischio minimo" che a loro volta sono da privilegiare rispetto agli applicativi "semplici da migrare". I software che non rientrano in nessuno dei livelli precedenti sono da considerare per ultimi.

Identificare per ogni software presente nella lista il livello di priorità. Tale livello può essere identificato sulla base dei principi specificati nelle sezioni seguenti ed utilizzando le domande proposte come esempi. Si raccomanda di considerare i molteplici aspetti che le domande fanno emergere nell'insieme per valutare l'appartenenza al livello.

Procedere, per uno specifico applicativo, in modo sequenziale partendo dal livello di priorità più alta ("opportunità da cogliere") fino a quello a priorità più bassa ("altro"). Se il software può essere considerato per il livello di priorità in esame, si passa all'applicativo successivo ripartendo dal livello a priorità più alta.

Un applicativo può logicamente ricadere in più livelli: va associato solo al livello di priorità più alta tra quelli applicabili.

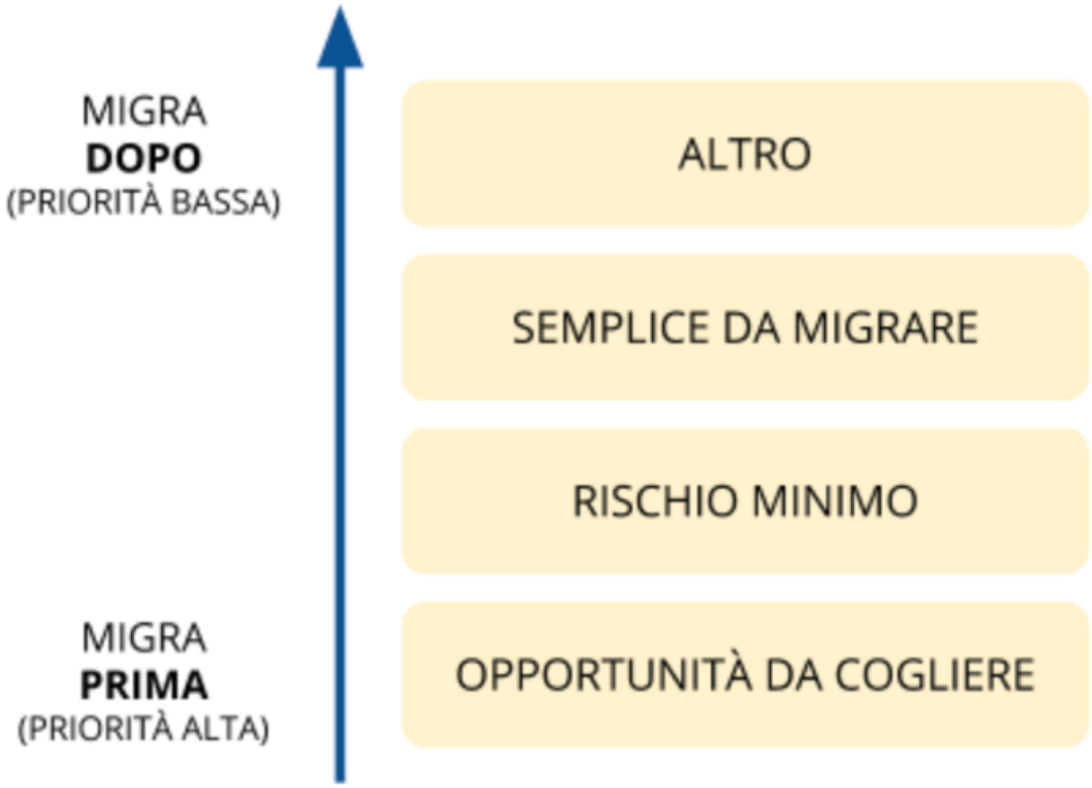
Una volta completata la classificazione degli applicativi sui quattro livelli, procedere con lo step successivo per gli applicativi appartenenti al livello di priorità più alto (non necessariamente in modo contemporaneo).

In caso ad un livello appartengano un numero significativo di applicativi è raccomandato di iterare la prioritizzazione utilizzando le dimensioni a priorità inferiore, ad es. se il livello "opportunità da cogliere" ha decine di applicativi, si può raffinare la prioritizzazione considerando per ognuno il livello di rischio, identificando quelli a rischio minimo. Se necessario, si può ulteriormente raffinare dando priorità, tra quelli con opportunità da cogliere e rischio minimo, a quelli più facili da migrare.

3.1.3.1 Livello 1: opportunità da cogliere

Gli applicativi che si consiglia di approfondire per primi per la migrazione sono quelli che a oggi hanno maggiori opportunità di trarre vantaggio (soprattutto in termini di costi) dal cloud.

[28] https://docs.italia.it/italia/developers-italia/lg-acquisizione-e-riuso-software-per-pa-docs/it/bozza/attachments/allegato-d-guida-alle-licenze-open-source.html


MIGRA
DOPO
(PRIORITÀ BASSA)
MIGRA
PRIMA
(PRIORITÀ ALTA)
ALTRO
SEMPLICE DA MIGRARE
RISCHIO MINIMO
OPPORTUNITÀ DA COGLIERE

Ecco alcune domande da porsi per identificare gli applicativi appartenenti a questo livello:

- Si prevedono significativi risparmi di costi con la migrazione al cloud di questo applicativo? Ad es.
  - La licenza software è in scadenza?
  - Si può risparmiare sulle spese per le strutture, l'alimentazione ed il raffreddamento?
  - Si può risparmiare sui costi di connettività?
- È necessaria una soluzione di disaster recovery?
- Adotta già una soluzione di disaster recovery onerosa?
- Questo applicativo richiede un aggiornamento hardware imminente che rende più interessante il passaggio al cloud prima piuttosto che più avanti nel tempo?
- Questo applicativo richiede un incremento delle risorse hardware?
- Questo applicativo richiede frequente manutenzione hardware?
- Ci sono applicativi nel cloud (soluzioni Saas) che renderebbero questa applicazione notevolmente migliore?
- Ci sono requisiti di conformità normativa per questa applicazione non ancora soddisfatti che possono essere risolti sul cloud?

Identificare questi applicativi, primi candidati per la migrazione, permetterà all'amministrazione di ottenere successi rapidi che producono vantaggi tangibili e immediati per gli utenti e l'organizzazione stessa.

3.1.3.2 Livello 2: ridurre al minimo il rischio di migrazione

Laddove il primo livello si concentra sulle opportunità, il secondo livello si concentra sul rischio. Quali applicazioni puoi spostare con un rischio relativamente basso per la continuità del servizio? Ci sono una serie di domande che l'IT può farsi per aiutare a valutare quali applicazioni sono meno rischiose da migrare, ovvero tra le più interessanti da migrare nelle prime fasi di un progetto di migrazione cloud. Per esempio:

- Qual è la criticità di questa applicazione per l'organizzazione? Qual è la sensibilità rispetto ai tempi di inattività? molto importante, 24x7 mission-critical? moderatamente importante? bassa importanza, ambiente dev / test? Guida: gli applicativi con minore criticità espongono ad un rischio minore
- Un alto numero di dipendenti e/o cittadini dipendono da questa applicazione? Guida: un minor numero di utilizzatori rappresenta un rischio minore
- Qual è il livello dell'ambiente di questa applicazione (produzione, staging, test, sviluppo)? Guida: gli ambienti non di produzione hanno un rischio minore
- Quante dipendenze e/o integrazioni non interoperabili ha questa applicazione (ovvero che non utilizzano API)? Guida: dipendenze/integrazioni basate su API rappresentano un rischio minore
- Qual è la conoscenza del team IT di questa applicazione? Guida: maggiore è la conoscenza, minore è il rischio
- Il team IT ha una documentazione completa e aggiornata per questa applicazione e la sua architettura? Diagramma di sistema, diagramma di rete, diagramma del flusso di dati, documentazione sulla build/deploy, documentazione della manutenzione in corso, .. Guida: più completa ed aggiornata è la documentazione, minore è il rischio
- Quali sono i requisiti di conformità normativa per questa applicazione? Guida: maggiori requisiti di conformità introducono più variabili da controllare, aumentando il rischio
- Qual è la sensibilità ai tempi di fermo e / o di risposta per questa applicazione? Guida: garantire tempi di risposta molto ridotti in specifici contesti possono rappresentare un rischio maggiore. Impatto elevato in caso di tempi di fermo rappresenta un rischio maggiore.
- Ci sono responsabili d'area desiderosi e disposti a migrare i loro applicativi in anticipo?

Porsi delle domande come quelle in elenco aiuta a classificare le applicazioni dal rischio più basso al più alto. Le applicazioni a basso rischio dovrebbero essere migrate per prime e le applicazioni a rischio più elevato dovrebbero invece essere migrate più tardi.

3.1.3.3 Livello 3: facilità di migrazione al cloud pubblico

Il terzo livello in questo framework ruota attorno alla facilità con cui è possibile migrare potenzialmente un'applicazione al cloud. A differenza del rischio, che riguarda l'importanza relativa di tale applicazione, la facilità di migrazione riguarda il modo in cui il trasferimento dell'applicazione verso il cloud sarà privo di attriti. Alcune buone domande da porsi includono:

- Come è stata sviluppata questa applicazione? Acquisto di terze parti da un produttore rilevante (ancora in attività?), acquisto di terze parti da un produttore minore (ancora in attività?), scritto in-house (autore ancora in organizzazione?), scritto da un partner (ancora attivo? Ancora un partner?)
- Quanto è nuova questa applicazione? È stata progettata per l'esecuzione on-premise o nel cloud? Adotta microservizi? È multi-tier?
- È possibile migrare questa applicazione utilizzando approcci semplici come lift-and-shift (re-host)? Utilizza macchine virtuali o container?
- Questa applicazione è strettamente dipendente da uno specifico sistema operativo o è flessibile rispetto a questo aspetto?
- Questa applicazione (o i suoi dati) ha requisiti normativi, di conformità per l'esecuzione on-premise? Guida: la conformità può aumentare la complessità della migrazione
- Quali sono le considerazioni sui dati per questa app? Sono aggiornati di frequente? Ci sono altri sistemi dipendenti da questo set di dati?

Quando si pianificano le applicazioni da migrare nel cloud, è possibile che a volte applicazioni di Livello 3 possano andare prima del Livello 2 (o anche Livello 1). Questo è assolutamente normale. Livello 2 e Livello 3 implicano molte variabili, quindi è comune avere un po" di scambi lungo il percorso di migrazione mantenendo comunque il senso logico della sequenza

3.1.3.4 Livello 4: altro

Il quarto ed ultimo livello di questo framework raccoglie tutti quegli applicativi che non hanno un evidente beneficio dalla migrazione al cloud, rappresentano un rischio significativo nella migrazione per i servizi che supportano e hanno una complessità specifica nella migrazione.

Questo tipo di applicativi sono tipicamente applicativi molto personalizzati o costituiti da soluzioni ad hoc per necessità particolari, per cui la loro migrazione pone sfide che altri applicativi di mercato non pongono e per i quali non ci si può affidare a conoscenza diffusa sul mercato.

Questi applicativi possono essere lasciati in fondo al processo di migrazione perché la combinazione dei fattori li rende meno appetibili dal punto di vista del valore generato rispetto agli altri e la complessità della migrazione richiede un'esperienza consolidata che si può avere dopo aver completato con successo le migrazioni precedenti.

4.2 3.2 Scheda di assessment dell'applicativo

Una volta identificati gli applicativi prioritari candidati alla migrazione in cloud, si devono ora valutare in dettaglio gli aspetti e le caratteristiche di ciascuno di essi per validarne l'opportunità rispetto al rischio e poi scegliere la strategia di migrazione più adatta (terzo step del processo presentato nel capitolo 2.3).

In particolare, gli obiettivi di questa seconda fase del processo sono:

- ricostruire una conoscenza di base sugli applicativi che sono stati prioritizzati
- evidenziare le informazioni utili (sia in ambito tecnico che di business) a supportare l'identificazione delle strategie di migrazione applicabili a ciascun applicativo
- stimolare la comunicazione fra personale tecnico e non-tecnico per la decisione su quale strategia di migrazione sia più adatta a ciascun applicativo prioritizzato

Per facilitare il raggiungimento di questi risultati, abbiamo creato una scheda di assessment dell'applicativo (vedi allegato "Scheda di assessment dell'applicativo in formato ODS" o "Scheda di assessment dell'applicativo in formato XLSX") che l'amministrazione può compilare seguendo le istruzioni riportate di seguito (ed esplicitate anche direttamente sul foglio di lavoro).

In generale, la scheda di assessment dell'applicativo è stata pensata per:

- bilanciare sforzo e valore con un modello strutturato ma al contempo snello
- stimolare valutazioni non abituali da fare in modo collaborativo

Le informazioni richieste non sono obbligatorie, ma la loro specificazione aiuta a prendere decisioni più consapevoli. Le informazioni temporaneamente non disponibili o di difficile recupero possono essere tralasciate. I dati reali sono da privilegiare rispetto alle stime se il loro recupero è sufficientemente agevole, altrimenti stime basate sull'esperienza ed il buon senso possono assolvere al medesimo obiettivo. Lo scopo non è la precisione e l'esaustività ma evidenziare le tendenze e gli aspetti più critici.

Si consiglia di avvalersi del supporto e dell'esperienza dell'unità di esecuzione e del centro di competenza per raccogliere le informazioni migliori.

Vediamo ora insieme le diverse sezioni che compongono la scheda di assessment e come compilarle.

4.2.1 3.2.1 Aspetti tecnologici

Gli aspetti tecnologici sono quelle caratteristiche che distinguono l'applicativo da un punto di vista tecnico e sui quali l'amministrazione dovrà focalizzarsi per valutare come migrare l'applicativo. I campi da riempire in questa sezione sono:

- **Stack tecnologico**: componenti tecnologiche attualmente in uso, ad es. database usato e rispettiva versione, ambiente di runtime, sistemi di notifica, sistemi di gestione di code, web server
- **Uso di componenti sostituibili con l'equivalente servizio cloud-native**: componenti on-premise utilizzate dall'applicativo che sono sostituibili con servizi gestiti dal cloud provider, ad es. database, LDAP, load balancer, SMTP server
- **Dimensionamento delle componenti infrastrutturali**: componenti infrastrutturali attualmente in uso che garantiscono il corretto funzionamento dell'applicativo, ad es. # di server, dimensione dello storage, CPU, memoria (questa informazione aiuta in particolare a calcolare l'impatto economico che la migrazione al cloud dell'applicativo può avere)
- **Utilizzo effettivo delle componenti infrastrutturali**: utilizzo effettivo delle risorse sulla base di misurazioni effettive o stime (questa informazione aiuta in particolare a dimensionare il cloud di destinazione e a calcolare l'impatto economico che la migrazione al cloud dell'applicativo può avere)
- **Dipendenza dall'hardware fisico**: situazione attuale in cui l'applicativo si trova in termini di dipendenza dall'hardware fisico, specificando ad esempio se si usano macchine virtuali, container oppure se l'applicativo o parti di esso sono su macchine fisiche
- **Misure di sicurezza**: minacce a cui l'applicativo è esposto, specificando quali componenti e dati sono più vulnerabili e richiedono protezione

- **Sistemi on-premise da cui dipende**: sistemi on-premise interni ed esterni da cui l'applicativo dipende, ad es. IDP, LDAP, sistema di notifiche. Non devono essere inclusi qui i sistemi già in cloud, come le piattaforme abilitanti della PA (ad es. Spid, PagoPA)
- **Sistemi on-premise che dipendono**: sistemi on-premise interni ed esterni che dipendono dell'applicativo, ad es. i sistemi che consumano i dati dell'applicativo

4.2.2 3.2.2 Vincoli tecnologici

I vincoli tecnologici sono quegli aspetti tecnologici particolarmente critici che hanno un ruolo stringente sulla scelta della strategia di migrazione e sui quali l'amministrazione dovrà focalizzarsi per decidere come effettivamente migrare l'applicazione. I campi da riempire in questa sezione sono:

- **Presenza di test di validazione**: presenza di test che possano validare le eventuali modifiche da effettuare sul codice sorgente per ridurre il rischio di regressione (ad es. test unitari, di integrazione, funzionali, performance). Per un approfondimento su questa tipologia di test vedi capitoli 5.5 e 6.4
- **Modificabilità del codice sorgente**: livello di modificabilità del codice sorgente, che può essere:
  - **nulla**, ovvero inesistente
  - **parziale**, se il codice può essere parzialmente modificato ad es. influenzando le scelte evolutive del prodotto realizzato da terze parti
  - **completa**, se il codice sorgente può essere completamente modificato ad es. in quanto se ne ha la proprietà o se la licenza del codice sorgente lo permette (ad es. open source)
- **Disponibilità di documentazione tecnica**: disponibilità della documentazione tecnica che spiega il funzionamento interno dell'applicativo e delle sue componenti per intervenire in modo mirato e controllato. Per approfondimento sulla documentazione vedi capitolo 4.3.2.2
- **Connettività minima necessaria**: impatto della connettività in termini di latenza ed ampiezza di banda, in relazione agli SLA che il fornitore deve garantire, sull'usabilità dell'applicativo. Si richiede di scegliere tra:
  - **rete locale**, se l'accesso alla rete locale è un requisito vincolante all'usabilità di questo applicativo
  - **internet**, se per l'applicativo l'accesso a internet è un requisito vincolante all'usabilità di questo applicativo

4.2.3 3.2.3 Dati

Ai dati è dedicata una sezione a parte in quanto rappresentano un aspetto molto importante nel contesto di una migrazione (per approfondimento sul tema vedi capitolo 6). I campi da riempire in questa sezione sono:

- **Dimensione della base di dati**: dimensione dei dati da migrare nell'unità di misura opportuna fra Byte, KB, MB, GB, TB, ecc.
- **Frequenza di consultazione dei dati (annuale)**: ogni quanto i dati vengono consultati nell'arco di un anno
- **Frequenza di aggiornamento dei dati (annuale)**: ogni quanto i dati vengono aggiornati nell'arco di un anno
- **Ciclo di vita dei dati**: quantità di tempo dopo la quale il dato può essere considerato obsoleto e quindi eliminabile dal sistema. Seguire le indicazioni di data retention del GDPR laddove applicabili. Si consiglia qui, se ritenuto necessario, di differenziare per tipologia di dato
- **Applicativi che trattano gli stessi dati**: applicativi che trattano gli stessi dati gestiti da questo applicativo (tutti o un sottoinsieme). In particolare, si fa riferimento a quegli applicativi che hanno una copia dell'insieme di dati considerato e che necessitano una sincronizzazione. Si consiglia di riportare in questo campo sia i nomi degli applicativi che il sottoinsieme dei dati trattati

4.2.4 3.2.4 Parti interessate

Le parti interessate sono tutte quelle persone e processi che, per diverse ragioni, sono interessate nella migrazione dell'applicativo. Alle persone l'amministrazione dovrà far riferimento per sapere chi contattare nel caso di domande specifiche o considerazioni da fare sulla migrazione dell'applicativo. Il campo da riempire in questa sezione è:

- **Rappresentanti delle aree impattate**: persone da coinvolgere o da tenere informate sia perchè con potere decisionale, sia perché utilizzatrici dell'applicativo o perché impattate dalla migrazione. Si raccomanda di considerare un eventuale coinvolgimento di personale esterno all'amministrazione (ad es. fornitori con un'influenza sulla migrazione). Nella scheda di assessment è presente un foglio di lavoro separato da riempire con queste specifiche informazioni
- **Processi impattati e punti di attenzione**: Riportare i processi interni o esterni dell'organizzazione che vengono impattati da questo applicativo e se vi sono dei punti d'attenzione da considerare durante il processo di migrazione

4.2.5 3.2.5 Bisogni

I bisogni sono quelle informazioni che identificano l'utilizzo effettivo dell'applicativo e le necessità che deve supportare. Considerare i bisogni servirà all'amministrazione per valutare l'opportunità connessa alla migrazione dell'applicativo. I campi da riempire in questa sezione sono:

- **# medio di utenti unici giornalieri negli ultimi 12 mesi**: numero medio di utenti unici in un giorno nell'ultimo anno. Il periodo considerato di 12 mesi vuole evitare periodi di prolungato inutilizzo. Per questo campo si consiglia di utilizzare, se disponibili, i dati degli strumenti di analytics
- **# massimo di utenti unici giornalieri negli ultimi 12 mesi**: numero massimo di utenti unici in un giorno nell'ultimo anno. Il periodo considerato di 12 mesi vuole evitare periodi di prolungato inutilizzo. Per questo campo si consiglia di utilizzare, se disponibili, i dati degli strumenti di analytics
- **# minimo di utenti unici giornalieri negli ultimi 12 mesi**: numero minimo di utenti unici in un giorno nell'ultimo anno. Il periodo considerato di 12 mesi vuole evitare periodi di prolungato inutilizzo. Per questo campo si consiglia di utilizzare, se disponibili, i dati degli strumenti di analytics. Se ci sono giorni in cui l'applicativo è inutilizzato o spento, mettere "0" come valore
- **Periodi di utilizzo in una settimana**: fasce orarie in cui il servizio è utilizzato durante una settimana. Devono essere evidenziate qui eventuali fasce in cui si hanno picchi di utilizzo significativo. Se l'utilizzo è mediamente costante nell'arco della giornata e della settimana, indicare «utilizzo omogeneo»
- **Periodi di utilizzo in un mese**: fasce orarie in cui il servizio è utilizzato durante il mese. Devono essere evidenziate qui eventuali fasce in cui si hanno picchi di utilizzo significativo. Se l'utilizzo è mediamente costante nell'arco del mese, indicare «utilizzo omogeneo»
- **Periodi di utilizzo in un anno**: fasce orarie in cui il servizio è utilizzato durante l'anno. Devono essere evidenziate qui eventuali fasce in cui si hanno picchi di utilizzo significativo. Se l'utilizzo è mediamente costante nell'arco dell'anno, indicare «utilizzo omogeneo»
- **Costi dell'infrastruttura**: tempi e costi per l'allestimento, la manutenzione dell'infrastruttura attuale ed il suo eventuale potenziamento (provisioning di nuove risorse). Questa informazione aiuta in particolare a calcolare l'impatto economico che la migrazione al cloud dell'applicativo può avere.
- **Licenze**: tutte licenze che sono utilizzate, specificando il loro costo e quando scadono. Anche questa informazione aiuta in particolare a calcolare l'impatto economico che la migrazione al cloud dell'applicativo può avere. Nella scheda di assessment è presente un foglio di lavoro separato da riempire con queste specifiche informazioni. Per il riuso considerare le Linee Guida su acquisizione e riuso di software per le pubbliche amministrazioni pubblicate su Docs Italia[29]

[29] https://docs.italia.it/italia/developers-italia/lg-acquisizione-e-riuso-software-per-pa-docs/it/stabile/

- **Criticità**: eventuali aspetti critici dell'applicativo, ad esempio:
  - performance o stabilità che impattano l'operatività degli utenti finali o richiedono una spesa specifica per la loro risoluzione temporanea (perché una definitiva non è attualmente possibile)
  - conformità normativa, ad esempio GDPR
  - sicurezza
- **Evoluzione del servizio nei prossimi tre anni**: aree di evoluzione previste o ipotizzate per il servizio supportato dall'applicativo per identificarne la centralità rispetto alla strategia dell'organizzazione. Tenere in considerazione, nel caso siano disponibili, i piani pluriennali definiti ed eventuali scadenze già definite. Considerare qui la tipologia e la numerosità delle evoluzioni attese per l'applicativo

4.2.6 3.2.6 Mercato

Le informazioni riguardo al mercato aiutano ad esplorare le opportunità presenti sul mercato per una migrazione al cloud dell'applicativo. I campi da riempire in questa sezione sono:

- **Alternative SaaS**: esistenza di alternative SaaS per l'applicativo in analisi all'interno del Cloud Marketplace[30] di ACN
- **Disponibilità di import dei dati**: garanzia che il fornitore SaaS provveda la possibilità di importare i dati all'interno del servizio SaaS tramite formati pubblici e aperti

4.3 3.3 Analisi costi-benefici

Determinare costi e benefici del cloud richiede un approccio sistematico che tenga in considerazione tutti i fattori diretti ed indiretti che impattano sulla migrazione.

Per un'analisi efficace è opportuno seguire questi passi:

1. definizione del periodo temporale su cui calcolare il ritorno sull'investimento (tipicamente 5 anni)
2. verifica dei costi attuali dell'infrastruttura e loro proiezione sul periodo temporale
3. stima dei costi dell'infrastruttura cloud e loro proiezione sul periodo temporale
4. stima dei costi di migrazione
5. stima dei costi di post-migrazione
6. valutazione dei costi rispetto ai benefici tangibili ed intangibili

Calcolare il costo totale di un servizio IT e compararlo con il potenziale ritorno economico di una migrazione al cloud costituisce un compito arduo, come lo è l'analisi dei costi sul cloud computing. Un'analisi dettagliata dei benefici di una migrazione deve includere valutazioni a breve, medio e lungo termine oltre ai costi di terminazione. A questo riguardo, ci sono due indicatori chiave da considerare:

Total Cost of Ownership (TCO) = Costi iniziali + Costi ricorrenti + Costi di terminazione

Ritorno sull'investimento (ROI) = ((Benefici tangibili + Benefici intangibili) - TCO) / TCO

I costi nascosti che le organizzazioni potrebbero avere difficoltà a rilevare in anticipo includono:

- il costo per un cambio di provider dovuto ad una variazione regolatoria o di linea di condotta: per calcolare gli investimenti necessari per cambiare piattaforma di cloud, nel caso in cui problemi economici o regolatori lo rendano necessario, le organizzazioni devono tener conto di diversi fattori, tra cui il pagamento per l'estrazione e la validazione dei dati ed il costo di assunzione di risorse IT necessarie per compiere questo lavoro

[30] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud

- le spese inaspettate dovute all'iniziale migrazione di sistemi: le amministrazioni pubbliche o le software house, infatti, possono dover riscrivere le applicazioni per operare in un ambiente virtualizzato e riformattare i dati per adattarli ai formati del SaaS del fornitore
- il lock-in con uno specifico modello di servizio proprietario: costi dovuti al fatto che l'amministrazione non riesce a svincolarsi facilmente da una scelta tecnologica precedentemente effettuata (vedi capitolo 4.3) o debba modificare gli applicativi perché i cloud provider dismette o modifica le feature esposte dal servizio acquistato

In generale, è importante considerare a livello economico le diverse opzioni di migrazione al cloud che si possono avere in base alle caratteristiche di ciò che si vuole migrare: retain, retire, re-purchase, re-host, re-platform o re-architect (vedi capitolo 4.1 per approfondire le strategie di migrazione). Ogni alternativa dovrebbe essere analizzata in dettaglio per decidere il modello cloud migliore in base al contesto e alle circostanze in cui l'amministrazione si trova.

4.3.1 3.3.1 Verifica dei costi attuali dell'infrastruttura

Per calcolare i costi attuali dell'infrastruttura on-premise è necessario considerare i suoi costi complessivi di utilizzo e manutezione nel tempo. Questo calcolo include i costi in capitale ed operativi, sia direttamente imputabili alla piattaforma che condivisi con altre aree.

I **costi diretti** imputabili alla piattaforma sono solitamente documentati in fatture e pagamenti. Tra questi:

- investimenti per l'acquisto iniziale di hardware (server fisici, dischi, apparati di rete, . . .) e servizi di sviluppo software
- costi operativi per la gestione del quotidiano, inclusa la forza lavoro impiegata nell'infrastruttura:
  - gestione e manutenzione dell'hardware e del software, acquisto di componenti danneggiate, ..
  - noleggio macchinari e hardware
  - connettività internet e networking dedicata al progetto, content delivery network, . . .
  - acquisti e rinnovi delle licenze delle componenti infrastrutturali come sistemi operativi, macchine virtuali, database, antivirus, backup, ..
  - acquisti e rinnovi delle licenze degli applicativi (CRM, collaboration suite, software sviluppati ad hoc, ..)
  - manutenzione delle soluzioni di data protection e disaster recovery (DAT, dispositivi di archiviazione sostitutiva, infrastrutture replicate, . . .)
  - formazione ICT

Ci sono poi i **costi indiretti**, ossia condivisi con altre strutture o progetti e che solitamente vengono ripartiti tra diverse aree o dipartimenti:

- manutenzione delle strutture di hosting, come gli immobili ed il personale necessario alla loro gestione
- manutenzione dell'infrastruttura di alimentazione (gruppi continuità, generatori, quadri comandi, ecc.) e di raffreddamento
- costi di connettività ad internet condivisi con altre aree/dipartimenti (tipo connessione, banda minima garantita, fallback in caso di fallimento, ecc.)
- costi amministrativi necessari per mantenere e amministrare il dipartimento IT. Questi possono includere le risorse da altri dipartimenti del proprio ente - personale, acquisti, ragioneria, ecc. - che sono dedicati a gestire lo staff IT interno ed esterno

A questi si aggiungono una serie di costi «nascosti» che possono cubare una fetta importante dei costi complessivi dell'IT, inclusi:

- i costi legati al rischio di mancata adeguatezza delle soluzioni on-premise alle evoluzioni normative o ai requisiti sicurezza

- la mancata produttività causata dall'indisponibilità dell'infrastruttura IT che è direttamente proporzionale al numero di utenti interni o esterni
- i costi legati a infortuni connessi alla manutenzione delle infrastrutture fisiche.

4.3.2 3.3.2 Stima dei costi dell'infrastruttura cloud

Partendo dai costi attuali dell'infrastruttura on-premise e dalla loro ripartizione, vanno stimati i costi dell'infrastruttura cloud. Questo lavoro è in parte semplificato dal fatto che molti cloud provider pubblicano online dei calcolatori utili a fare delle stime - che solitamente sono indicative e non sostituiscono un preventivo vero e proprio.

Alcuni di questi calcolatori includono anche i presunti costi dell'on-premise come la forza lavoro impiegata o l'energia elettrica e permettono di scaricare delle analisi comparative tra cloud e on-premise: anche se la valutazione dei costi dell'infrastruttura attuale non può essere demandata ai calcolatori esposti dai CSP, confrontare i valori ed i riparti ritornati con quelli preventivamente stimati può essere utile.

Utilizzare i calcolatori dei prezzi dei CSP qualificati da ACN nel Cloud Marketplace, laddove disponibili.

Nel caso più semplice di uno IaaS, dove basta approvvigionarsi di server e storage, possiamo stimare il costo di una infrastruttura cloud analoga inserendo in un generico calcolatore queste informazioni:

- **server**:
  - tipo di server
  - numero di macchine virtuali
  - core della CPU
  - memoria in GB
  - hypervisor, sistema operativo guest e motore DB, se si immette un tipo di server
- **storage**:
  - tipo di storage
  - capacità di archiviazione grezza (raw)
  - percentuale di storage ad accesso sporadico

Calcolatori più avanzati permettono di dettagliare meglio server e storage differenziando per tipologia di carico (eg. ETL, HPC, DB, . . .) oppure includono i servizi SaaS e PaaS come:

- database as a service
- servizi di machine learning
- identity e account management.

4.3.3 3.3.3 Stima dei costi di migrazione al cloud

Il passo successivo è la stima dei costi da sostenere per la migrazione degli applicativi nel cloud. Vanno considerati i costi di:

- **spostamento dei dati nel cloud**: è un passaggio cruciale. A parte i costi che i CSP solitamente addebitano per il consumo di banda, sono da considerare costi per manodopera e automazione necessaria a sincronizzare i dati dell'infrastruttura attuale con quella in cloud. E' necessaria una conversione degli schemi o dei formati? Quanto si perde nella conversione? Vanno scritte procedure di allineamento per far coesistere per il tempo necessario l'infrastruttura esistente e quella in cloud? Queste sono solo alcune delle domande che possono aiutare a stimare questi costi;

- **integrazione e test delle app**: deve essere possibile migrare le applicazioni su cloud e testare tutte le funzionalità.

  Non tutte le applicazioni sono pronte per il cloud. La migrazione di software legacy e mainframe, sistemi ERP (enterprise resource planning[31]), applicazioni nate per l'on-premise è un processo complesso, difficilmente testabile e a volte impraticabile. Bisogna mappare le interazioni delle applicazioni con tutte le componenti delle infrastrutture, individuare le modifiche necessarie al funzionamento sul cloud e quelle necessarie per il testing di tutte le funzionalità a valle della migrazione.

- **spese di consulenza**: se l'organizzazione non dispone di tutte le competenze necessarie per eseguire autonomamente una migrazione al cloud può attingere a competenze esterne, ad esempio per sviluppare una strategia, pianificare un'architettura cloud, o eseguire il processo di migrazione.

  Le spese di consulenze sono determinate dalle competenze necessarie, dal numero di esperti da coinvolgere e dalla durata della migrazione; per stimarle serve aver chiari i criteri di selezione per i consulenti, conoscere approfonditamente quali sono le competenze ed i punti di forza e di debolezza dell'organizzazione in materia di cloud.

- **licenze:** è importante eseguire una valutazione dei costi-benefici associati alla migrazione in cloud di software on-premise sotto licenza. Per informazioni più dettagliate si rimanda al capitolo 3.4 Gestione delle licenze software in cloud.

Ricordiamo che il framework di lavoro del programma di abilitazione al Cloud delle PA prevede centri di competenza sul territorio, ovvero dei soggetti aggregatori di tecnici, esperti e managers dell'IT per consolidare e potenziare le competenze, il *know how* e l'esperienza relativa alla gestione dei servizi cloud nelle amministrazioni. Questi centri saranno il punto di riferimento per le pubbliche amministrazioni che si apprestano ad iniziare il proprio percorso verso il cloud.

4.3.4 3.3.4 Stima dei costi post-migrazione

Che cosa si deve pagare dopo aver completato la migrazione al cloud? I costi di infrastruttura mensili che sono stati calcolati nel secondo passaggio di analisi (vedi sezione 3.3.2), ovviamente.

Tuttavia, è necessario tenere in considerazione anche i costi diretti e indiretti necessari per mantenere e migliorare il nuovo ambiente cloud, in quanto molti di questi continueranno a essere pagati anche dopo il completamento della migrazione iniziale.

In questa voce ricadono, tra gli altri, i costi di:

- evoluzione delle operation, come l'aumento dello spazio dedicato al monitoraggio delle risorse, l'acquisto di nuovi servizi necessari alla crescita della piattaforma o degli strumenti di CI/CD;
- gestione delle pratiche amministrative e legali per assicurare il rinnovo dei contratti o la conformità alle ultime normative sulla data protection;
- monitoraggio dei livelli di servizio concordati e delle soluzioni di sicurezza adottate dal CSP;
- formazione necessaria per far operare il personale su infrastrutture più moderne e articolate.

4.3.5 3.3.5 Valutazione dei costi rispetto ai benefici tangibili ed intangibili

Dopo aver calcolato tutti i costi, si potrebbe arrivare ad un numero elevato rispetto a quanto si pensava o ad eventuali costi attuali (tipicamente solo diretti) che si hanno in mente. Eppure è probabile che quel numero sia più piccolo di tutti i costi che si stanno attualmente pagando per l'infrastruttura on-premise.

Ma oltre ai risparmi sui costi, il cloud porta anche un elevato numero di benefici immateriali che possono essere difficili da misurare direttamente. Consente ad un'organizzazione di essere più flessibile e agile in modo da poter

[31] https://it.wikipedia.org/wiki/Enterprise_resource_planning

testare e lanciare i servizi più velocemente e reagire meglio alle mutevoli condizioni del mercato. Non ci si deve più preoccupare di acquistare e configurare nuovi server per gestire la domanda elevata, dato che è possibile scalare automaticamente i server cloud istantaneamente. E si ha la tranquillità che la probabilità di un down degli applicativi è minima grazie all'elevata disponibilità, al bilanciamento del carico e alle funzionalità di backup dei fornitori cloud.

Alcuni di questi benefici sono già stati trattati nel capitolo 1.2, ma approfondiamo qui quelli da tenere in particolare considerazione durante l'esecuzione di un'analisi costi-benefici.

3.3.5.1 Differenziale dei costi sul cloud rispetto ai costi on-premise

Confrontando i valori dei costi sul cloud e dei costi on-premise sul periodo considerato, si può identificare il beneficio tangibile creato dall' eliminazione dei canoni di manutenzione richiesti dall'hardware di proprietà e dei periodici acquisti per il rinnovo degli asset, dallo snellimento delle attività sia tecniche (verifica funzionamento, segnalazione malfunzionamenti, verifica apparecchiature obsolete) che amministrative (gare, impegni di spesa, liquidazioni fatture, ecc.), dalla riduzione dei costi di energia elettrica e tutte le altre voci impattate dalla migrazione.

3.3.5.2 Dimensionamento reale o elasticità reale

Le soluzioni on premise sono tipicamente dimensionate rispetto alla capacità necessaria per gestire il massimo carico previsto, sia esso dovuto ad una crescita del servizio o a situazioni temporanee di picco. Il provisioning delle macchine virtuali, della banda, della memoria e della CPU o della spazio di storage sono dimensionati sulla base di questi valori massimi che si prevedono di dover gestire.

Questo è legato al fatto che le infrastrutture on-premise sono poco elastiche, ovvero risulta complesso aumentare o diminuirne il dimensionamento: i tempi per aumentare le risorse a disposizione sono significativi ed una volta acquisite nuove risorse non è tipicamente vantaggioso rilasciarle, in particolare se solo per un periodo. Questo rende l'infrastruttura on premise non dimensionata sul bisogno attuale.

Grazie alla facilità ed alla rapidità di allocazione di nuove risorse su una piattaforma cloud, il dimensionamento deve essere effettuato sulle correnti necessità, aumentando o diminuendo le risorse allocate solo in caso di necessità.

Analizzare l'utilizzo effettivo delle risorse è quindi cruciale per un corretto dimensionamento della soluzione in cloud. Per questo tipo di analisi consultare metriche di utilizzo o utilizzare strumenti di mercato che forniscono questo tipo di analisi.

3.3.5.3 Riduzione dei rischi di disservizio operativo, perdita dati e del rischio reputazionale

Gli applicativi in cloud godono di alta disponibilità, ovvero la probabilità che i servizi siano indisponibili per problemi infrastrutturali è molto bassa. Grazie alla possibilità di fare provisioning delle risorse in tempi molto rapidi è anche possibile rispondere a situazioni di carico non previste in modo tempestivo. Ciò impatta il rischio di disservizio con i costi che questo ha associati.

Il rischio di perdita di dati per problemi infrastrutturali come la rottura di un dispositivo sono altresì praticamente inesistenti, azzerando i costi, tipicamente molto ingenti, legati alla perdita di dati.

Grazie ai servizi di backup e ripristino disponibili in cloud è anche possibile ritornare ad una situazione funzionante con minima perdita di dati in tempi molto rapidi, nel caso vi siano motivi applicativi o di violazione dei sistemi di sicurezza che causano una perdita di dati.

Il rischio reputazionale per l'ente causato dai problemi sopra elencati ed il costo ad esso associato, anche se di difficile quantificazione economica ma tipicamente elevato nel tempo, è quindi anch'esso ridotto significativamente.

3.3.5.4 Semplificazione del disaster recovery

L'allestimento di un sito di disaster recovery in cloud è molto semplice ed i suoi costi sono legati al suo utilizzo effettivo. In base all'architettura dell'applicativo in cloud, ridondato su più data center, tale sistema potrebbe diventare implicito.

3.3.5.5 Disponibilità di aggiornamenti, bugfix e miglioramenti più rapida

Il passaggio in cloud permette aggiornamenti dell'applicativo più rapidi e questo impatta le attività rendendo sempre disponibile la versione più aggiornata ed affidabile dell'applicativo senza costi per l'organizzazione.

Può essere utile valutare anche l'impatto economico di problemi verificatisi in passato a causa di mancata tempestività nella risoluzione o opportunità non colte in passato per il medesimo motivo.

3.3.5.6 Adeguamenti normativi su sicurezza e privacy

Amministrare le infrastrutture IT comporta responsabilità di sicurezza e di protezione dei dati personali. Le recenti normative in materia di privacy e di sicurezza informatica impongono anche alle pubbliche amministrazioni l'adozione di misure tecniche e organizzative adeguate a garantire la sicurezza del trattamento dei dati.

Molti provider di servizi cloud offrono un'ampia gamma di criteri, tecnologie e controlli che rafforzano la sicurezza complessiva, grazie alla protezione dei dati (che possono essere criptati con i più alti livelli di sicurezza del mercato), dell'applicazione e dell' infrastruttura da minacce potenziali.

Questo permette agli enti di utilizzare soluzioni complete, già mature e disponibili o, a volte, trarne vantaggio in modo del tutto trasparente in quanto soluzioni applicate in modo totalmente trasparente dal cloud provider, senza dover investire soluzioni ad hoc e nelle competenze necessarie per capire di quello di cui si necessita.

3.3.5.7 Miglioramento del servizio (percezione dell'utente finale)

Sfruttando le potenzialità del cloud, le pubbliche amministrazioni hanno l'opportunità di migliorare la qualità dei propri servizi, siano questi ad uso interno o ad uso del cittadino.

Grazie al cloud, l'amministrazione può gestire i servizi in maniera più efficiente ed efficace, riuscendo a concentrarsi maggiormente sulle funzionalità da offrire ai propri utenti. Questo ha un ritorno economico in termini di efficacia, efficienza e reputazione dei servizi.

4.4 3.4 Migrazione delle licenze software in cloud

Le licenze sono un aspetto da considerare con attenzione nella migrazione al cloud in quanto riguardano diversi ambiti: sistemi operativi, application server, database, strumenti e molto altro.

Gli accordi di licenza definiscono come il software di un fornitore può essere utilizzato da un'organizzazione. La maggior parte di questi però è stata scritta quando gli ambienti IT tipici erano costituiti da PC desktop e server fisici di vari tipi, tutti di proprietà e gestiti dal cliente. I data center disponevano di rack pieni di server per singole applicazioni, ognuna con il proprio storage collegato. Negli ultimi due decenni la sala server si è però evoluta attraverso la virtualizzazione dei server stessi, fino alla virtualizzazione dell'intera infrastruttura di calcolo. Allo stesso modo, l'utilizzo dei servizi da parte degli utenti finali si è evoluto passando dalla fruizione su un singolo dispositivo per utente a potenzialmente più dispositivi (laptop, desktop, tablet, dispositivo mobile) per ogni utente.

Gli accordi di licenza a loro volta sono cambiati per tenere conto di questo panorama in evoluzione. Tramite i diritti di utilizzo secondari ad esempio, si consente l'uso del software in maniera non simultanea su un desktop e un laptop, o, con la trasformazione sempre più verso abbonamenti per utente, si consente l'implementazione su dispositivi illimitati.

Nel mondo dei data center c'è stata un'evoluzione, per tenere conto delle macchine virtuali e quindi dei processori ad alta densità di core. Gran parte del software server side è ora concesso in licenza "per core" o su base utente nominativa. In alcuni casi è stato anche interrotto il collegamento tra l'hardware fisico sottostante e il software eseguito su di esso. È stato concesso il diritto di spostare il software in modo dinamico, solitamente a un costo aggiuntivo o con alcune restrizioni ed alcuni fornitori stanno iniziando ad affrontare le sfide di licenza presentate dalla containerizzazione.

I modelli di licenza del software sono dunque in costante evoluzione, al passo con il corrispondente panorama tecnologico e, come parte integrante del piano di migrazione al cloud, è compito delle organizzazioni verificare se le licenze utilizzate on-premise possano essere trasferite o convertite in licenze cloud o meno.

Abbiamo illustrato le diverse tipologie di licenze on-premise e open source nel capitolo 3.1. Di seguito parleremo delle tipologie di licenze cloud e di alcuni accorgimenti necessari quando si considera la migrazione di un applicativo sotto licenza da on-premise al cloud.

4.4.1 3.4.1 Tipologie di licenze cloud

Vi sono diverse tipologie di licenze cloud:

- **licenza ad abbonamento**: permette di pagare per l'utilizzo del software o servizio per un determinato periodo di tempo con anche la possibilità di annullare in qualsiasi momento l'abbonamento.Questo tipo di licenze vengono solitamente fornite su periodi mensili o annuali. Pagando mensilmente si ha una maggiore flessibilità potendo scegliere se mantenere o meno il servizio attivo. Il pagamento di una licenza annuale può però portare a significativi risparmi sui costi. Molti fornitori consentono inoltre di iniziare con una licenza mensile per poi passare a una licenza annuale proporzionale, fornendo così una sorta di periodo di prova. Questo tipo di licenza è molto comune tra i fornitori di servizi SaaS.
- **licenza per utilizzo** (**pay-as-you-go**): la tecnologia ha permesso ai fornitori di avere accesso a strumenti avanzati che gli permettono di monitorare accuratamente l'utilizzo dei loro software e servizi, aprendo la porta a un diverso tipo di licenza: per utilizzo o pay-as-you-go. Con questo tipo di licenza si paga sulla base della quantità del sistema che si utilizza. Vi sono diverse modalità di misurazione del consumo, come ad esempio:
  - Numero di processi eseguiti sul server del fornitore
  - Quantità di spazio su disco utilizzato
  - Dimensione del database e / o numero di query effettuate

Questo tipo di licenza consente di iniziare con il numero di risorse necessarie e di scalare con la crescita con l'assunzione che se viene usato maggiormente il software o l'hardware, è perché si sta crescendo e l'investimento maggiore è quindi giustificato. Un ultimo aspetto di questa licenza è che consente ai venditori di monitorare l'utilizzo su tutti i dispositivi, eliminando il problema, sia per il cliente che per il fornitore, di provare a concedere in licenza il software su ciascun dispositivo separatamente.

- **licenza per istanza** (**pay-by-instances**): questo modello di licenza si applica principalmente ai servizi cloud IaaS e PaaS. In questo scenario, si paga per ogni server o istanza di macchina virtuale che il fornitore esegue. Questo modello di licenza offre molti dei vantaggi della licenza per utilizzo perché si paga solo ciò di cui hai bisogno e si utilizza. Risulta un modello di licenza molto conveniente soprattutto quando si debbono creare istanze per provare qualcosa o mostrare dei prototipi per i quali non sarà poi più necessario mantenere attiva l'istanza.
- **Bring Your Own License** (**BYOL**): come visto in precedenza, vi sono varie licenze utilizzabili on-premise e una cosa in comune a tutte queste è il concetto di proprietà dell'hardware su cui vengono fatti girare gli applicativi. In cloud però non si possiede l'hardware, non si ha idea su quale server sia in esecuzione il software licenziato ed esistono dinamiche come la capacità di assegnare nuove risorse hardware in modo dinamico in caso di traffico in eccesso. Vi è quindi una nuova esigenza relativa a come assicurarsi di rispettare i requisiti di licenza relativi a territorio, calcolo e proprietà nel cloud. La risposta dei venditori è stata (in alcuni casi) consentire l'utilizzo delle licenze on-premise su infrastruttura cloud dedicata come istanze dedicate, host dedicati e istanze riservate. Le istanze dedicate possono essere utilizzate laddove la licenza consenta l'utilizzo per macchina virtuale mentre

gli host dedicati forniscono la conformità per il software concesso in licenza per host fisico. Le istanze riservate possono essere utilizzate per ridurre potenzialmente i costi nel tempo - offrono infatti uno sconto in cambio di un impegno a lungo termine (come 12 mesi). Poiché il server è dedicato, viene considerato come un'estensione del data center on-premise dal punto di vista delle licenze. Sebbene questo tipo licenze possano consentire l'utilizzo di software esistenti on-premise in cloud, è solitamente necessario pagare un costo aggiuntivo. Oltre a questo costo è importante considerare anche l'aspetto relativo al rischio. La proprietà delle licenze infatti rimane e con essa la responsabilità di garantire la conformità in caso di processi di audit. Ad esempio, è necessario assicurarsi che le licenze precedentemente utilizzate on-premise siano ora utilizzate solo nel cloud.

4.4.2 3.4.2 Gestione delle licenze per la migrazione al cloud

La gestione delle licenze in una prospettiva che privilegi la migrazione al cloud deve seguire questi approcci:

- Le licenze in scadenza vanno considerate con priorità per la migrazione ad una soluzione cloud-based (applicativo "opportunità da cogliere") e le licenze mantenute attive solo il tempo minimo necessario a completare la migrazione
- Le licenze ad uso perpetuo vanno considerate per BYOL se possibile o per la dismissione ed il passaggio ad una soluzione cloud con la relativa licenza. In caso vi siano investimenti significativi in ammortamento è necessario definire un termine per la dismissione della licenza e pianificare la migrazione in modo che il sistema in cloud sia disponibile e pienamente operativo per quel termine
- I software middleware[32] on-premise sotto licenza come ad esempio gli application server o DBMS sono tipicamente sfruttati da più di un applicativo. Questi componenti non vengono migrati immediatamente in quanto sono necessari a tutti quegli applicativi ancora on-premise e in attesa di migrazione.

  Il risultato è una situazione ibrida in cui è necessario mantenere attivi sia i middleware on-premise che i middleware, o loro alternative, in cloud.

  Oltre che l'impatto sulla gestione, questa duplicazione di ambienti anche se temporanea comporta un costo relativo alle licenze per le quali è dunque importante considerare bene l'impatto sul costo della migrazione ed eventuali accordi possibili con i fornitori di servizi su tipologie di licenze ibride.

[32] https://it.wikipedia.org/wiki/Middleware

CAPITOLO 5

4. Pianificare la migrazione

Nel capitolo 3 abbiamo illustrato il processo da seguire per individuare gli applicativi più adatti per iniziare con la migrazione al cloud, ovvero quelli migrabili con maggiori benefici e minori criticità. Una volta individuati questi applicativi e approfonditi i loro aspetti con la compilazione della scheda di assessment, la pubblica amministrazione deve pianificare in che modo eseguire la loro migrazione.

In questo capitolo presentiamo le modalità con cui è possibile eseguire una migrazione (le sei strategie di migrazione) e per ciascuna di esse illustriamo benefici, rischi e criteri di applicabilità. L'obiettivo è aiutare l'amministrazione a scegliere la strategia di migrazione più adatta a ciascuno degli applicativi precedentemente individuati.

Inoltre, dedichiamo la parte centrale di questo capitolo alla valutazione delle competenze, asset fondamentale da considerare prima di iniziare con la migrazione al cloud.

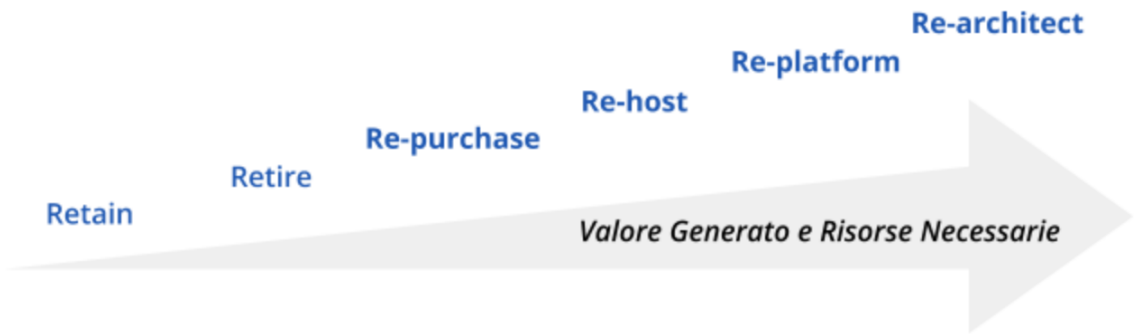
Infine, trattiamo qui degli aspetti chiave da considerare nelle fasi di scelta del cloud service provider e di progettazione dell'affidamento al fornitore che supporterà l'amministrazione nel processo di migrazione. A questo fanno riferimento le due sezioni conclusive di questo capitolo riguardo gli SLA e il tema del lock-in.

5.1 4.1 Le strategie di migrazione

Le principali strategie adottate per la migrazione di applicativi al cloud sono note come le 6R:

- Retain o Conservazione
- Retire o Smantellamento
- Re-purchase o Sostituzione
- Re-host o Trasferimento di host
- Re-platform o Trasferimento di piattaforma
- Re-architect o Rifattorizzazione/Creazione di una nuova architettura

Ogni strategia si differenzia dalle altre per valore generato e per risorse necessarie alla sua realizzazione. É quindi necessario identificare le strategie applicabili alla migrazione di un applicativo in base al livello di ambizione cui si



aspira ed alle effettive risorse che si possono utilizzare: non esiste *la strategia corretta* ma il giusto bilanciamento di queste due componenti in base al contesto in cui si opera.

Il valore generato da una migrazione può essere espresso in termini di risparmio sui costi dell'infrastruttura e della sua gestione, agilità nelle operazioni di configurazione dei servizi, velocità di approvvigionamento, tempestività di adattamento al variare dei bisogni reali del servizio (scalabilità delle risorse in base al traffico, dimensionamento in base al numero di operatori effettivi o al consumo necessario per le esigenze di business).

In base al valore che si vuole ottenere, è altresì richiesto un livello direttamente proporzionale di risorse in termini temporali, finanziari e di competenze supportati da un approccio strategico adeguato.

In questo capitolo descriveremo in dettaglio ogni strategia di migrazione, fornendo per ciascuna di esse:

- definizione
- benefici e rischi connessi
- criteri di applicabilità
- indicazione dei campi specifici e dei loro valori all'interno della scheda di assessment (vedi capitolo 3.2), che identificano la strategia come applicabile ad un determinato applicativo

È infine importante ricordare che per ogni applicativo spesso è possibile individuare più di una strategia di migrazione: l'obiettivo di questa fase è di identificare lo spettro delle strategie possibili. La scelta finale su quale strategia utilizzare dipenderà da diversi fattori quali la capacità di investimento sul processo, il valore generato, le tempistiche, le competenze e tutte quelle considerazioni da fare negli step successivi del processo (ovvero con l'analisi costi-benefici e la valutazione delle competenze) come illustrato nel capitolo 2.3.

5.1.1 4.1.1 Conservazione o Retain

La strategia di conservazione o retain consiste nel prendere la decisione consapevole di non migrare in cloud un determinato applicativo e di mantenerlo attivo sulla propria infrastruttura on-premise definendo una nuova data in cui rivalutare i fattori che hanno portato a questa decisione e nel caso non fossero più attuali procedere con la migrazione in cloud dell'applicativo tramite una delle altre strategie presentate in questo capitolo.

Questa decisione può essere guidata da diversi fattori: strategici, di incompatibilità con le attuali piattaforme cloud o di assenza di software alternativi in SaaS (vedi capitolo 1.1).

Criteri di applicabilità della strategia

Le principali caratteristiche che permettono di identificare gli applicativi candidati alla strategia di retain o conservazione sono:

- forte dipendenza dalla connettività che può portare ad un degrado eccessivo dell'esperienza utente nella fruizione da cloud
- recente acquisizione con quota di ammortamento importante riferita ad un investimento in corso
- basso impatto sulla strategia digitale dell'amministrazione non ne giustificano un redesign completo o parziale (Re-architect)
- forte obsolescenza tecnologica non supportata da cloud service provider che rende il trasferimento di hosting (Re-host) o piattaforma (Re-platform) non possibili
- assenza di un'alternativa SaaS che soddisfi le richieste dell'amministrazione
- requisiti di sicurezza e privacy dei dati personali per cui l'amministrazione preferisce mantenere la parte dei dati soggetta a tali requisiti on-premise e trasferire in cloud solo la parte con minori vincoli

Caratteristiche peculiari sulla scheda di assessment dell'applicativo

Rispetto alla scheda di assessment dell'applicativo compilata nella fase di assessment vi sono determinate caratteristiche che rendono un servizio candidabile per questa strategia

- Connettività minima necessaria = rete locale
- Evoluzione del servizio nei prossimi 3 anni per valutare se l'impatto sulla strategia dell'amministrazione è basso
- Stack tecnologico per valutare l'obsolescenza tecnologica
- Alternative SaaS = nessuna
- Misure di sicurezza per valutare se come amministrazione si ritiene ci siano motivazioni valide per la conservazione on-premise
- Licenze per valutare se l'investimento su una licenza valida on-premise non sia recuperabile o trasferibile con una corrispondente licenza in cloud

5.1.2 4.1.2 Smantellamento o Retire

La strategia di smantellamento o retire consiste nell'identificare gli applicativi che non sono più utili e possono essere spenti per focalizzare l'attenzione sulle risorse che sono maggiormente utilizzate.

Sono da considerarsi come "non più in uso" anche quegli applicativi il cui beneficio per l'amministrazione pubblica è inferiore ai costi complessivi di mantenimento ed il cui utilizzo è limitato ad un insieme predefinito e ricorrente di funzionalità con bassa frequenza.

Per questi la strategia di smantellamento deve definire, per le necessità ancora presenti e non più coperte dalle funzionalità dell'applicativo da dismettere, le modalità di espletamento dopo lo smantellamento, definendo anche approcci manuali o semi-manuali.

La dismissione di un applicativo necessita di una verifica sugli utilizzi da parte di operatori o dipendenze verso altri sistemi non noti ma in realtà ancora attivi. Per questa verifica è possibile prevedere un periodo significativo di spegnimento del sistema con possibilità di riaccensione, in cui monitorare eventuali segnalazioni di malfunzionamenti o di impossibilità a completare attività.

Benefici

- eliminazione dell'infrastruttura a supporto dell'applicativo e delle sue attività di gestione
- eliminazione della necessità di ruoli di supporto specializzato su applicativi legacy[33]
- focalizzazione dell'attenzione sulle risorse che sono maggiormente utilizzate

Rischi

[33] https://it.wikipedia.org/wiki/Sistema_legacy

- adattamento dei processi interni

Criteri di applicabilità della strategia

Le principali caratteristiche che permettono di identificare gli applicativi candidati alla strategia di retire o smantellamento sono:

- applicativi non più utili
- applicativi rimpiazzati da versioni più moderne e mantenuti attivi al solo scopo di rendere possibile l'accesso a dati storici che non sono stati migrati ai nuovi sistemi
- applicativi che non generano più nuovi dati
- applicativi utilizzati per creare report su dati storici
- applicativi la cui dismissione è stata procrastinata in quanto precedentemente non vi erano le condizioni per procedere, ad es. in termini di tempo, budget o competenze

Caratteristiche peculiari sulla scheda di assessment dell'applicativo

Rispetto al scheda di assessment dell'applicativo compilata nella fase di assessment vi sono determinate caratteristiche che rendono un servizio candidabile per questa strategia

- Frequenza di consultazione dei dati
- Se la frequenza di aggiornamento dei dati è zero ed il ciclo di vita si è concluso, è possibile smantellare rimuovendo anche i dati
- Se la frequenza di aggiornamento dei dati è zero ma quella di consultazione non lo è, allora è possibile smantellare l'applicativo migrando i dati in cloud
- Evoluzione del servizio nei prossimi 3 anni per valutare se l'impatto sulla strategia dell'amministrazione è basso
- Applicativi che trattano gli stessi dati per valutare se qualche applicativo tratta lo stesso insieme di dati e quindi può essere considerato come sostituto
- Periodi di utilizzo, # medio di utenti, # massimo di utenti, # minimo di utenti unici giornalieri negli ultimi 12 mesi per valutare se l'utilizzo è assente o non rilevante

5.1.3 4.1.3 Sostituzione o Re-purchase

I prodotti software sono fruibili principalmente attraverso due modalità:

1. on-premise: scaricando, installando e configurando il pacchetto software sulla propria infrastruttura
2. come Software as a Service (SaaS), ovvero accedendo ed utilizzando direttamente il prodotto, tipicamente attraverso un browser o un'app, senza alcuna responsabilità infrastrutturale o manutentiva

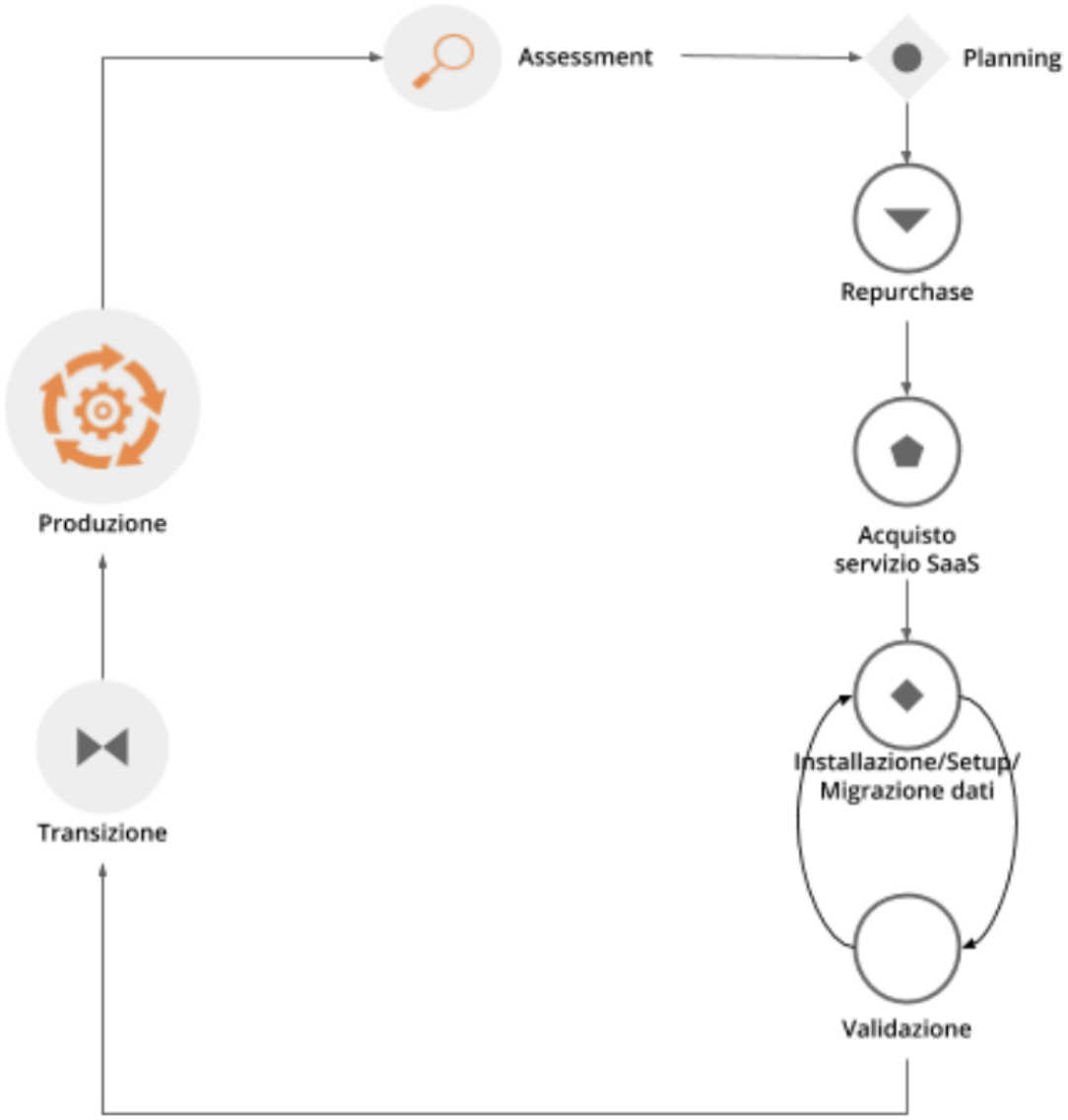
La strategia di Re-purchase consiste nel **rimpiazzare un applicativo installato e gestito on-premise con la controparte SaaS**.

Possiamo rappresentare questa strategia con il seguente diagramma:

Se un applicativo on-premise non ha un'alternativa SaaS fornita dal medesimo produttore, bisogna procedere con un benchmark comparativo per identificare un'alternativa secondo le linee guida su acquisizione e riuso di software per le pubbliche amministrazioni[34] e in ogni caso adottando il principio "Cloud First"[35].

Applicativi in modalità SaaS offrono modelli di pricing generalmente basati su sottoscrizioni invece di licenze come nel mondo on-premise: non è possibile l'acquisto di licenze perpetue ma sono diffuse sottoscrizioni con cadenza mensile o annuale basate su utilizzatori o consumo effettivo.

[34] https://lg-acquisizione-e-riuso-software-per-la-pa.readthedocs.io/it/latest/
[35] https://buildmedia.readthedocs.org/media/pdf/cloud-italia-docs/latest/cloud-italia-docs.pdf


Assessment
Planning
Repurchase
Acquisto
servizio SaaS
Installazione/Setup/
Migrazione dati
Validazione
Transizione
Produzione

Esempi di servizi che è possibile migrare su SaaS:

- Posta elettronica
- File Server per la condivisione dei file
- ERP

Benefici

- eliminazione dell'infrastruttura a supporto dell'applicativo e delle sue attività di gestione
- minori costi iniziali
- tempo di fermo per gli aggiornamenti dei sistemi ridotto
- accessibili da qualunque device connesso ad internet
- facilmente e tempestivamente adattabile alle variazioni delle necessità di business, ad es. n. di operatori cui garantire l'accesso (per user), livello di funzionalità disponibili (feature based), quota di risorse (per storage/transaction)
- no setup, accesso immediato

Rischi

- lock-in con il cloud service provider (vedi capitolo 4.3)
- migrazione dati verso un modello dati differente
- migrazione dei punti di integrazione (es. API) verso contratti differenti
- integrazione/riconfigurazione dei servizi SaaS con servizi on-premise (es. autentica)
- assenza di controllo in caso di down
- total cost of ownership (TCO)[36] potrebbe essere maggiore sul lungo periodo

Criteri di applicabilità della strategia

Identificazione della categoria dell'applicativo

La categoria di un applicativo riflette tipicamente il principale bisogno soddisfatto dalla sua applicazione. Non vi è una classificazione di riferimento, univoca e stringente delle soluzioni software, per cui l'identificazione deve utilizzare approssimazioni basate sui principali bisogni indirizzati, e sulle più diffuse classificazioni nel mercato. In un mercato ad alta innovazione è altresì possibile che nuovi prodotti definiscano nuove categorie che superano o specializzano quelle precedenti.

Esempi di categorie di software con opzioni SaaS disponibili sul catalogo dei servizi Cloud per la PA qualificati sono:

- CMS, Content Management Systems
- ERP, Enterprise Resource Planning
- LMS, Learning Management System
- Project Management
- Email Management Software
- File Sharing
- Collaborazione e Produttività
- Software amministrativi
- Gestione tributi

[36] https://it.wikipedia.org/wiki/Total_Cost_of_Ownership

Caratteristiche peculiari sulla scheda di assessment dell'applicativo

Rispetto al scheda di assessment dell'applicativo compilata nella fase di assessment vi sono determinate caratteristiche che rendono un servizio candidabile per questa strategia:

- Alternative SaaS
- Disponibilità di import dei dati

Lo sviluppo del mercato dei prodotti software verso la modalità SaaS, offre un costante aumento di soluzioni in cloud che possono rimpiazzare software precedentemente disponibile solo on-premise con la corrispondente versione cloud-based realizzata dal medesimo produttore o con soluzioni equivalenti o migliorative proposte da nuovi soggetti.

La verifica di tali alternative può essere fatta sul catalogo dei servizi cloud qualificati da ACN (Cloud Marketplace)[37], la piattaforma che espone i servizi e le infrastrutture qualificate.

All'interno del Cloud Marketplace è possibile ricercare i servizi e visualizzarne la scheda tecnica che mette in evidenza le caratteristiche tecniche, il modello di costo e i livelli di servizio dichiarati dal fornitore in sede di qualificazione.

A decorrere dal 1 aprile 2019, le pubbliche amministrazioni potranno acquisire esclusivamente servizi IaaS, PaaS e SaaS qualificati da ACN e pubblicati nel cloud Marketplace.

5.1.4 4.1.4 Trasferimento di host o Re-host

Anche detta strategia di *Lift and Shift*, consiste nel prendere (Lift) l'intero servizio, compreso di infrastruttura, architettura, dati e traffico e spostarlo su un hosting cloud (Shift) senza modifiche al core dell'applicativo. Spesso il re-host è una strategia che permette di fare un primo step verso il cloud valutando poi successivamente ulteriori miglioramenti all'applicativo che permettano di sfruttare ulteriormente i vantaggi del cloud.

Possiamo rappresentare questa strategia con il seguente diagramma:

La strategia può essere eseguita in due modi:

1. **automatizzata** tramite strumenti di migrazione
2. **manuale**

La migrazione automatizzata in presenza di strumenti di migrazione forniti dal cloud service provider e dai suoi partner è da considerarsi preferibile rispetto alla manuale perchè fornisce una strutturazione al processo, riduce le possibilità di errori, trae vantaggio dalle caratteristiche intrinseche della soluzione cloud cui si migra.

La strategia manuale è preferibile solo a fronte dell'impossibilità di procedere in modo automatico con strumenti maturi ed affidabili o a fronte di ben identificati obiettivi, come, ad esempio, permettere al team di familiarizzare con il cloud, migliorando così la propria competenza interna attraverso una delle strategie di migrazione più semplici.

Rehost automatizzato

I cloud service provider più diffusi forniscono la possibilità di migrare gli applicativi tramite l'utilizzo di strumenti dedicati, sviluppati internamente o forniti da partner esterni, che permettono di automatizzare l'intero processo di migrazione.

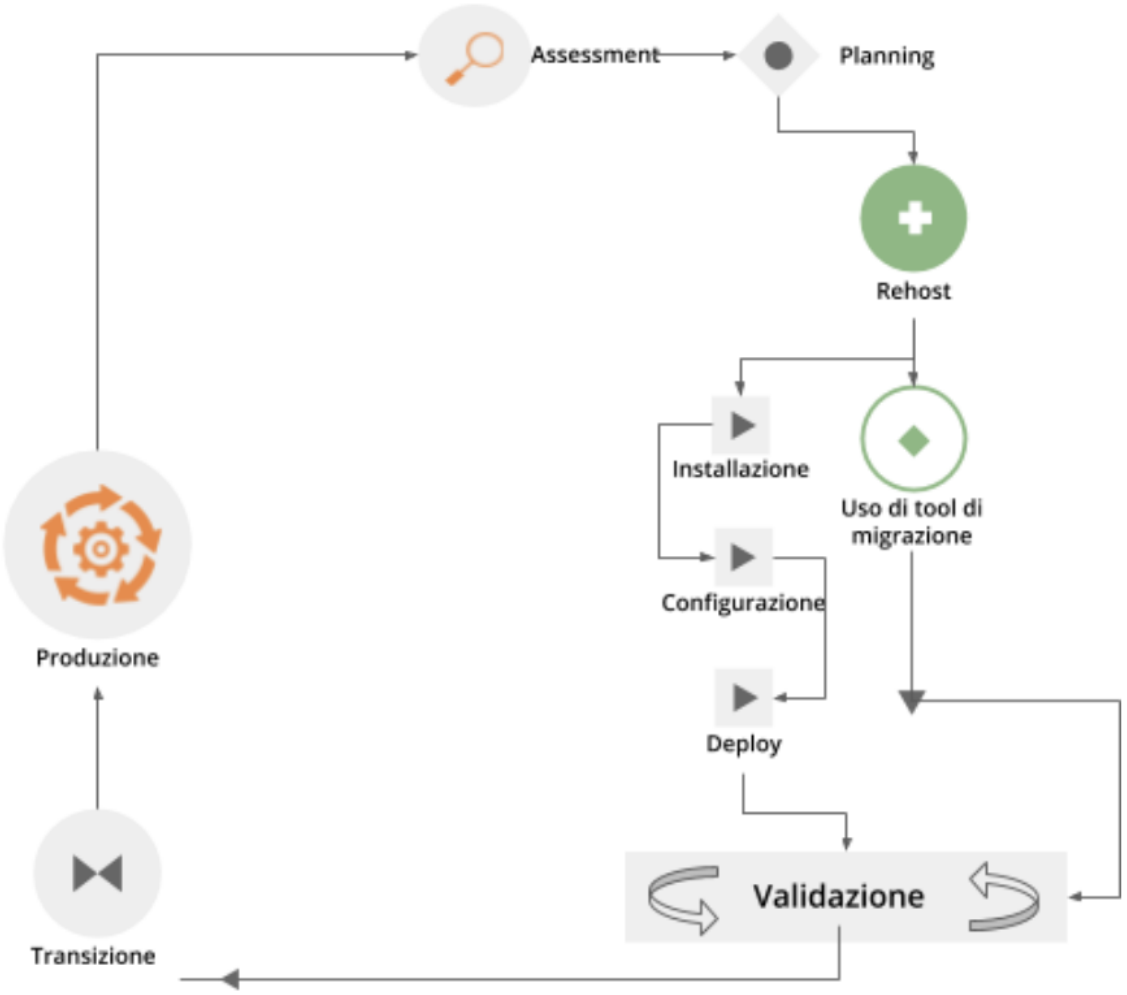
Questi strumenti sono solitamente divisi in 2 categorie:

1. **strumenti di migrazione dei server**: trasferiscono un'intera macchina fisica o virtuale su una corrispondente macchina in cloud
2. **strumenti di migrazione dei database**: trasferiscono i dati presenti da un database on-premise ad uno in cloud

Generalmente il processo associato all'utilizzo di questi strumenti rispecchia i seguenti passi:

1. **installazione** di un software agente sulla macchina o collegamento al database da migrare

[37] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud


Assessment
Planning
Rehost
Installazione
Uso di tool di migrazione
Configurazione
Deploy
Validazione
Produzione
Transizione

2. **definizione delle specifiche in cloud** della macchina o database di destinazione
3. **replica** della macchina o database, con entrambe le versioni funzionanti e dati sincronizzati
4. **testing** della replica, verificando che la macchina in cloud rifletta esattamente la sorgente migrata o che i dati su database siano corretti
5. **cut-over** dove gli utenti utilizzatori della macchina o database iniziale vengono reindirizzati verso quelli migrati in cloud

È sempre necessario fare riferimento alla documentazione dei singoli strumenti per ulteriori dettagli.

Rehost manuale

Ogni migrazione ha delle sue particolarità relative all'applicativo, l'infrastruttura, il team e l'organizzazione cui è applicata, ma possiamo generalizzare le procedure di rehosting manuale a questa serie di step:

1. **virtualizzazione** della macchina che ospita l'applicativo
2. **replica della macchina virtuale** sul nuovo servizio cloud
3. **sincronizzazione dei dati** tra Virtual Datacenter e cloud
4. **testing e validazione** del funzionamento del nuovo ambiente
5. **cut-off del traffico** e reindirizzamento verso il nuovo applicativo

Benefici

- ricchezza di strumenti di supporto: il mercato offre diversi strumenti per automatizzare il processo rendendolo più affidabile e solido
- riuso di competenze diffuse: le competenze sviluppate in ambito sistemistico e di gestione di infrastrutture virtualizzate on-premise sono sufficienti per poter procedere
- tempistiche di migrazione inferiori in media rispetto a re-purchasing, re-platforming e re-architecting
- riduzione delle risorse utilizzate a livello di infrastruttura e delle attività per la loro gestione
- dismissione delle risorse on-premise e costi associati
- maggiore possibilità di procedere con re-platform da un re-host grazie alla conoscenza della soluzione cloud acquisita durante la migrazione, la riduzione della complessità del sistema ottenuta dal passaggio da on-premise a cloud, la possibilità di creare ambienti di testing con effort molto ridotto

Rischi

- sovradimensionamento delle risorse: gli applicativi on-premise vengono solitamente dimensionati sulla base dei picchi di carico previsti, sovradimensionando quindi l'infrastruttura rispetto all'utilizzo abituale. Fare re-host senza riconsiderare il dimensionamento può comportare un'allocazione non necessaria di risorse con conseguente impatto sui costi.
- mancato sfruttamento di tutte le potenzialità del cloud: l'applicativo è migrato con la configurazione dell'ambiente on-premise e richiede una riconfigurazione degli aspetti di scalabilità orizzontale e verticale per sfruttare l'elasticità disponibile in un ambiente cloud. Il re-host deve essere principalmente solo un punto di partenza in una strategia evolutiva dell'applicativo/servizio che punti al raggiungimento di livelli di valore più alti.
- aumento della latenza dell'applicativo a causa di una differente connettività

Criteri di applicabilità della strategia

Le principali caratteristiche che permettono di identificare gli applicativi che possono adottare una strategia di migrazione in cloud di tipo re-host sono:

- applicativi con codice sorgente proprietario di terze parti che non hanno una roadmap evolutiva tendente al cloud in medio o breve termine

- soluzioni monolitiche per cui non è possibile sostituire le singole componenti applicative in una progressiva trasformazione finalizzata a sfruttare più propriamente le soluzioni cloud based
- soluzioni legacy basate su tecnologie obsolete
- applicativi con molte integrazioni con prodotti, servizi o librerie di terze parti
- alto impatto sulle risorse infrastrutturali
- soluzioni stagnanti che hanno raggiunto una stabilità evolutiva ed hanno una bassissima frequenza di aggiornamenti

Caratteristiche peculiari sulla scheda di assessment dell'applicativo

Rispetto al scheda di assessment dell'applicativo compilata nella fase di assessment vi sono determinate caratteristiche che rendono un servizio candidabile per questa strategia

- Modificabilità del codice sorgente = no
- Uso di componenti sostituibili con l'equivalente servizio cloud native = nessuno
- Stack tecnologico per valutare l'obsolescenza tecnologica
- Sistemi on premise da cui dipende per valutare la complessità generata dalle dipendenze e l'impatto sul processo di migrazione
- Sistemi on premise che dipendono per valutare la complessità generata dalle dipendenze e l'impatto sul processo di migrazione
- Connettività minima necessaria = internet
- Licenze per valutare se l'investimento su una licenza valida on-premise sia recuperabile o trasferibile con una corrispondente licenza in cloud
- Evoluzione del servizio nei prossimi 3 anni per valutare se l'impatto sulla strategia dell'amministrazione è basso

Queste caratteristiche evidenziano applicativi che risultano molto difficili da trasformare sia per possibilità tecnica che per rischio di efficacia, che hanno un impatto importante sull'attuale infrastruttura e che in generale possono essere migrati così come sono in quanto poco strategici nella roadmap futura.

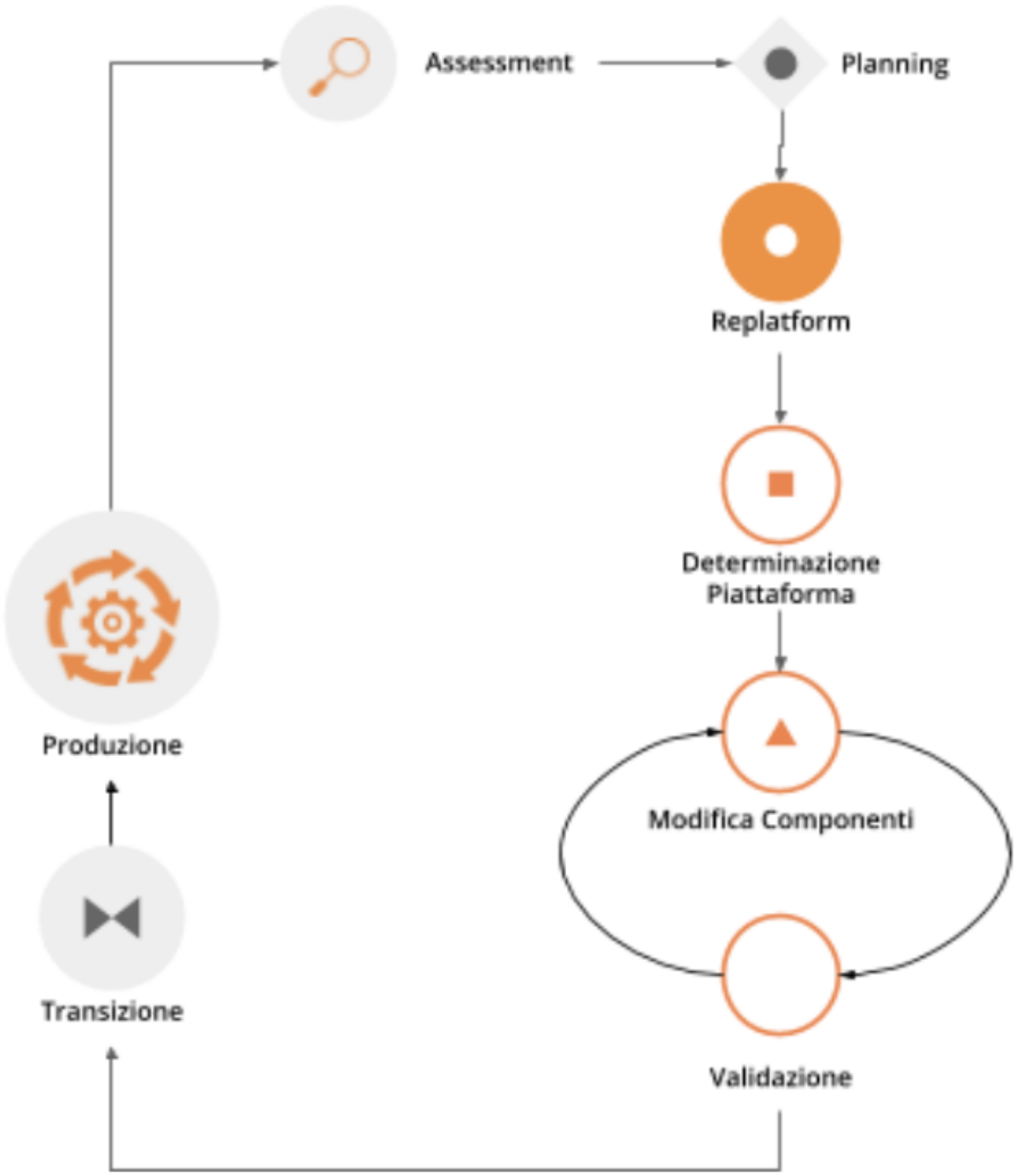
5.1.5 4.1.5 Trasferimento di piattaforma o Re-platform

La strategia di Re-platform oltre a trasferire un applicativo sul cloud come nel re-host, sostituisce nel processo di migrazione alcune componenti per meglio sfruttare le specificità della piattaforma di destinazione.

Possiamo rappresentare questa strategia con il seguente diagramma:

Esempi di sostituzione sono:

- i bilanciatori di carico che sui sistemi on-premise sono tipicamente macchine virtuali mentre in cloud sono disponibili come servizio gestito. Questa sostituzione permette di risparmiare sul numero di macchine virtuali e conseguentemente su costi e gestione
- il database management system on-premise con la versione completamente gestita in cloud per migliorare la resilienza della base dati sfruttando la disponibilità e semplicità di configurazione dei meccanismi di scalabilità, ridondanza, backup, patching, sicurezza, data encryption, hardware fault tolerance e monitoring
- l'ambiente di runtime attraverso l'utilizzo di container, ad es. Docker, per delegare la gestione di memoria, cpu e storage alla piattaforma cloud semplificando gli aspetti manutentivi e aumentando la portabilità fra diversi cloud service provider riducendo quindi il rischio di lock-in
- l'esecuzione dei batch demandata ai servizi specifici del cloud service provider per una riduzione dell'impatto manutentivo, del consumo di risorse ed una migliore scalabilità


Assessment
Planning
Replatform
Determinazione
Piattaforma
Modifica Componenti
Validazione
Produzione
Transizione

- lo storage per i file on-premise con l'equivalente servizio in cloud per sfruttare la capacità elastica, la robustezza, i meccanismi di sincronizzazione e gestione del ciclo di vita del dato e la modulazione dei costi in base alla frequenza di accesso al dato stesso di quest'ultimo

Benefici

- maggiore riduzione delle risorse utilizzate a livello di infrastruttura e delle attività per la loro gestione rispetto al re-host nel breve periodo
- migliore sfruttamento delle caratteristiche proprie del cloud come disponibilità, scalabilità, osservabilità, resilienza, provisioning delle risorse
- sviluppo di una conoscenza più profonda del cloud e dei servizi che offre senza modifiche radicali al software

Rischi

- difficoltà nel reperire le competenze necessarie per le trasformazioni che si vogliono operare, principalmente legate alla conoscenza dei sistemi in cloud e alle tecniche di refactoring.
- aumento del rischio di instabilità dell'applicativo in caso di trasformazioni multiple contemporanee: è altamente raccomandato di prioritizzare solo le trasformazioni che portano ad un beneficio tangibile ed applicarle in modo iterativo e controllato per validarne l'effetto.

Criteri di applicabilità della strategia

Le principali caratteristiche che permettono di identificare gli applicativi che possono adottare una strategia di migrazione in cloud di tipo re-platform sono:

- componenti separabili come ad esempio nelle architetture a 3 livelli o Three-tier dove logica di presentazione, logica di business e dato sono ben distinte
- utilizzo di servizi esterni sostituibili (ad esempio servizio SMTP o di autenticazione)
- necessità di migliorare la scalabilità
- frequenti modifiche

Caratteristiche sul scheda di assessment dell'applicativo

Rispetto al scheda di assessment dell'applicativo compilata nella fase di assessment vi sono determinate caratteristiche che rendono un servizio candidabile per questa strategia:

- Stack tecnologico per valutare architetture modulari e a componenti separabili
- Uso di componenti sostituibili con l'equivalente servizio cloud-native
- Periodi di utilizzo per valutarne la variabilità e confronto tra # medio di utenti e # massimo e minimo di utenti con l'obiettivo di identificare scostamenti rilevanti
- Utilizzo effettivo delle componenti infrastrutturali in confronto al dimensionamento delle componenti infrastrutturali per valutare un sovra o sotto dimensionamento
- Evoluzione del servizio nei prossimi 3 anni per valutarne l'importanza e l'opportunità di investimenti sull'applicativo
- Dipendenza dall'hardware fisico = se virtuale o container
- Connettività minima necessaria = internet
- Modificabilità del codice sorgente = parziale o completa
- Disponibilità di documentazione tecnica che supporti nella sostituzione delle componenti
- Criticità legate a componenti sostituibili con un'alternativa cloud native

Queste caratteristiche evidenziano un applicativo con un'architettura modulare, che utilizza componenti che possono essere sostituite con un equivalente servizio gestito dal cloud service provider, di cui si può modificare il codice sorgente per le parti di interfacciamento con tali componenti grazie anche alla conoscenza derivata dalla documentazione delle strutture interne.

Questi applicativi fanno parte della visione strategica dell'amministrazione che giustifica l'investimento nella trasformazione.

5.1.6 4.1.6 Rifattorizzazione/Creazione di una nuova architettura o Re-architect

La strategia di Re-architect ha come obiettivo quello di ripensare significativamente l'architettura core di un applicativo in ottica cloud, attraverso un processo di redesign iterativo ed incrementale che miri ad adottare appieno i servizi cloud-native offerti dai cloud service provider per massimizzare i benefici che ne derivano.

Esempi di redesign dell'architettura riguardano:

- l'adozione di *lambda-function* per scomporre un applicativo in modalità service-oriented sfruttando la capacità di autoscaling che dimensiona l'utilizzo sulla base del traffico effettivo
- l'utilizzo di *API gateway* per definire ed esporre interfacce applicative pubbliche o ad accesso controllato per favorire l'interoperabilità con sistemi esterni
- la trasformazione dell'applicativo in *componenti stateful e stateless*, ovvero con o senza stato interno persistente, per poter configurare lo scaling e l'availability in modo differenziato e sfruttare quindi in modo ottimale le risorse non essendo costretti ad un dimensionamento basato sul caso peggiore
- la creazione di un layer di integrazione che permetta di rimuovere la necessità di duplicazione dei dati tra applicativi diversi, consentendone il recupero direttamente dalla sorgente primaria

La strategia di re-architect, rispetto alle altre viste finora, permette di massimizzare lo sfruttamento delle potenzialità del cloud in termini di scalabilità, ridondanza, continuità del servizio, costi infrastrutturali e di gestione, ecc. Essa è al tempo stesso la più complessa da condurre in quanto richiede una conoscenza specialistica della piattaforma cloud utilizzata, ovvero principi di design cloud-native, metodologie consolidate di test coverage, test automation, refactoring o trasformazione del codice sorgente in modo controllato.

Possiamo rappresentare il re-architect con il seguente diagramma:

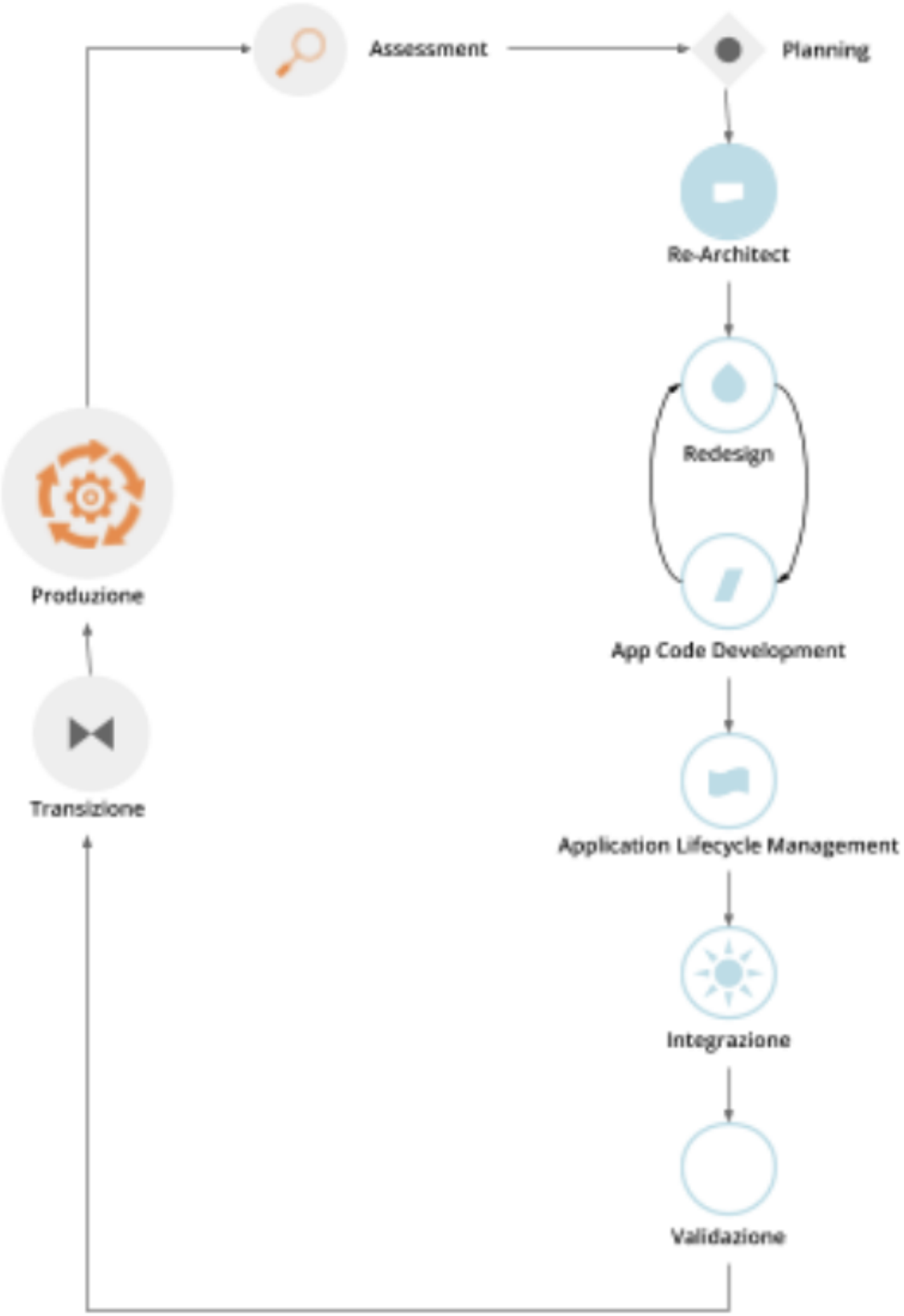
Benefici

- maggiore riduzione delle risorse utilizzate a livello di infrastruttura e delle attività per la loro gestione rispetto a re-host e re-platform nel breve e medio periodo
- ottimizzazione dei costi nel lungo termine grazie all'utilizzo delle risorse basato sull'effettiva necessità e non su quella prevista
- migliore sfruttamento delle caratteristiche proprie del cloud come disponibilità, scalabilità, osservabilità, resilienza, provisioning delle risorse
- miglioramento delle modalità di sviluppo e validazione attraverso strumenti avanzati per la sperimentazione come l'A/B testing[38] e deployment indipendenti delle componenti applicative
- responsività alle variazioni di carico impreviste grazie ad uno scaling in real time
- incremento della sicurezza grazie alla disponibilità di funzionalità avanzate

Rischi

- difficoltà nel reperire le competenze necessarie per le trasformazioni che si vogliono operare, principalmente legate alla conoscenza dei sistemi in cloud, tecniche di refactoring e principi di design di applicativi cloud native

[38] https://it.wikipedia.org/wiki/Test_A/B


Assessment
Planning
Re-Architect
Redesign
App Code Development
Application Lifecycle Management
Integrazione
Validazione
Transizione
Produzione

- aumento del rischio di instabilità dell'applicativo in caso di trasformazioni multiple contemporanee: è altamente raccomandato di prioritizzare solo le trasformazioni che portano ad un beneficio tangibile ed applicarle in modo iterativo e controllato per validarne l'effetto.
- rischio di significativo lock-in con il cloud service provider

Caratteristiche degli applicativi migrabili con re-architect

Di seguito una lista di caratteristiche che permettono di identificare gli applicativi la cui migrazione in cloud può essere preferibile con un approccio re-architect:

- centralità nella strategia di trasformazione digitale dell'ente
- necessità di un ammodernamento tecnologico e riduzione del debito tecnico per facilitare evoluzioni future
- bisogno di aumentare e ridurre la capacità di gestione del traffico per rispondere a necessità contingenti e variabili
- necessità di adeguamento alle linee guida del nuovo modello di interoperabilità del sistema informativo della PA

Caratteristiche sul scheda di assessment dell'applicativo

Rispetto al scheda di assessment dell'applicativo compilata nella fase di assessment vi sono determinate caratteristiche che rendono un servizio candidabile per questa strategia

- Evoluzione del servizio nei prossimi 3 anni per valutarne l'importanza e l'opportunità di investimenti sull'applicativo
- Stack tecnologico per valutare la necessità di ammodernamento
- Uso di componenti sostituibili con l'equivalente servizio cloud-native
- Criticità per identificare opportunità di miglioramento strutturale della soluzione
- Periodi di utilizzo per valutarne la variabilità e confronto tra # medio di utenti e # massimo e minimo di utenti con l'obiettivo di identificare scostamenti rilevanti
- Utilizzo effettivo delle componenti infrastrutturali in confronto al dimensionamento delle componenti infrastrutturali per valutare un sovra o sotto dimensionamento
- Connettività minima necessaria = internet
- Modificabilità del codice sorgente = parziale o completa
- Presenza di test di validazione per verificare il miglioramento apportato dalle modifiche intraprese e ridurre il rischio di regressione durante il processo
- Disponibilità di documentazione tecnica che supporti il processo di rifattorizzazione

Queste caratteristiche evidenziano un applicativo centrale per la visione strategica dell'amministrazione giustificandone l'investimento in tempo, competenze e costi per un redesign dell'architettura possibile grazie alla proprietà del codice sorgente o alla capacità di influenzare la roadmap evolutiva definita dal produttore.

5.2 4.2 Le competenze necessarie

Come abbiamo spiegato all'inizio di questo documento (vedi capitolo 2.3), iniziare una migrazione verso il cloud è un problema complesso i cui fattori sono correlati tra loro da relazioni non causali. Tutti i fattori devono perciò essere presi in considerazione prima di decidere da dove e come partire. A seconda del proprio contesto, delle proprie persone e delle loro competenze, ciascuna pubblica amministrazione dovrà valutare qual è il giusto punto di partenza bilanciando i diversi fattori e rimanendo pronta a reagire ed iterare a seconda dei risultati raggiunti.

Per questo motivo, in questo sottocapitolo forniamo uno strumento per abilitare la valutazione delle competenze interne alla pubblica amministrazione rispetto a quelle necessarie per eseguire una migrazione. L'obiettivo è quello di aiutare le amministrazioni ad individuare le competenze mancanti internamente ed avere chiarezza su cosa acquisire o cercare da eventuali fornitori che le supporteranno nella migrazione al cloud.

Prima di presentare lo strumento in dettaglio, descriviamo qui anche il processo con il quale esso è stato creato. In questo modo, infatti, l'amministrazione potrà scegliere se usare il framework fornito o replicare il processo per costruire uno strumento ancora più specifico e adatto al proprio contesto e alle proprie esigenze.

5.2.1 4.2.1 Definizione delle competenze necessarie

A seconda della strategia di migrazione scelta, saranno necessarie diverse competenze per poter portare a termine la migrazione con successo. Per valutare ed eventualmente acquisire le competenze necessarie per eseguire la migrazione, i passi da fare sono:

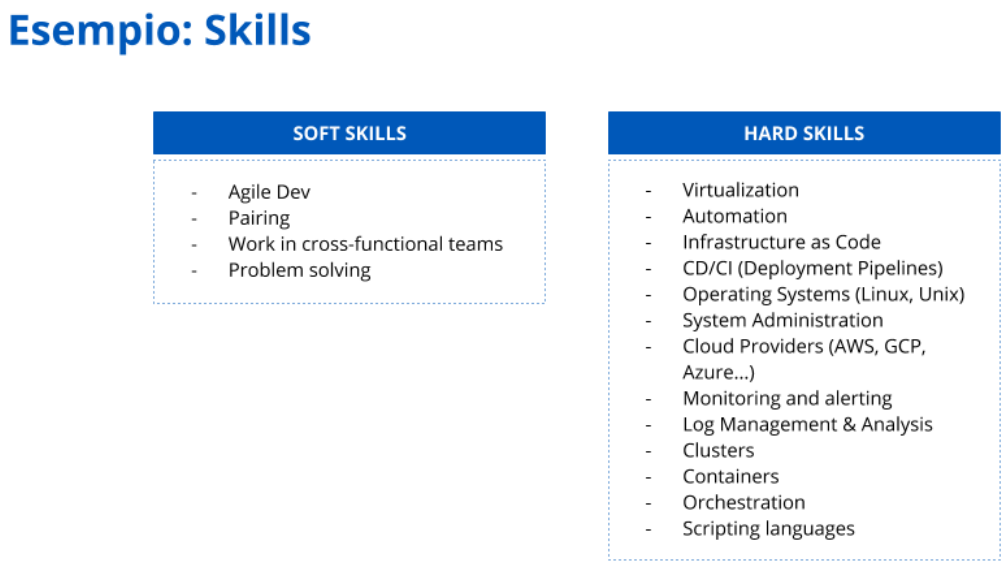
1. definizione delle competenze necessarie per la strategia scelta
2. valutazione interna delle competenze che il team dell'amministrazione già possiede
3. identificazione delle competenze da ricercare all'esterno tramite il coinvolgimento di fornitori o un'appropriata formazione del personale interno

Per quanto riguarda il primo passo, ovvero la definizione delle competenze necessarie, nella sezione 4.2.3 troverete uno strumento che presenta un set di competenze necessarie già definite che possono essere usate dall'amministrazione come punto di partenza per la valutazione.

Tuttavia, per poter definire le competenze necessarie è buona norma coinvolgere il team dell'amministrazione che sarà più vicino ai progetti di migrazione, in particolare quello tecnico, se presente. Con questo team, si consiglia di organizzare una o più sessioni in cui i partecipanti potranno contribuire in modo collaborativo allo scopo di definire:

1. **le aree di competenza necessarie**: macrocategorie che raggruppano una serie di skills, ad esempio come mostrato nell'immagine

2. **le skill per ognuna delle aree di competenza individuate**: divise in skill non-tecniche (ad es. di stampo metodologico e comunicativo) e tecniche, ad esempio come mostrato nell'immagine e che verranno poi distribuite nelle aree di competenza individuate



Una volta definite le aree di competenza e le skills che le compongono, il passo successivo consiste nell'effettuare una valutazione interna sulle singole skills. Questo può essere fatto mappando le competenze con un modello di maturità che permetta al team dell'amministrazione di evidenziare su quali ambiti sono già competenti e su quali invece sarà necessario un supporto esterno o una formazione specifica.

5.2.2 4.2.2 Mappare le competenze secondo un modello di maturità

Mappare le competenze interne con un modello di maturità aiuta ad ottenere un quadro realistico dei diversi livelli della loro padronanza da parte del team a oggi e facilita la pianificazione per il loro sviluppo o per la loro acquisizione, se totalmente assenti, in futuro.

Tornando al processo illustrato sopra, una volta identificate le aree di competenza e le rispettive skills necessarie, il team dell'amministrazione deve ora procedere alla valutazione di ciascuna skill secondo cinque livelli (basati sul modello di Dreyfus[39]):

1. **principiante**: se non si ha nessuna o pochissima conoscenza della competenza in questione e sono necessarie guida e istruzioni precise per essere produttivi con questa skill
2. **principiante avanzato**: se si ha una certa autonomia nell'applicazione della skill in maniera situazionale, ma sempre tramite il supporto di guida e istruzioni
3. **intermedio**: se si ha una conoscenza sufficientemente profonda delle regole associate alla skille e si possono raggiungere obiettivi concreti in maniera autonoma
4. **competente**: se si ha una visione d'insieme e si è in grado di fare scelte sulla base del contesto e dell'esperienza, le istruzioni sono usate come fallback

[39] https://apps.dtic.mil/dtic/tr/fulltext/u2/a084551.pdf

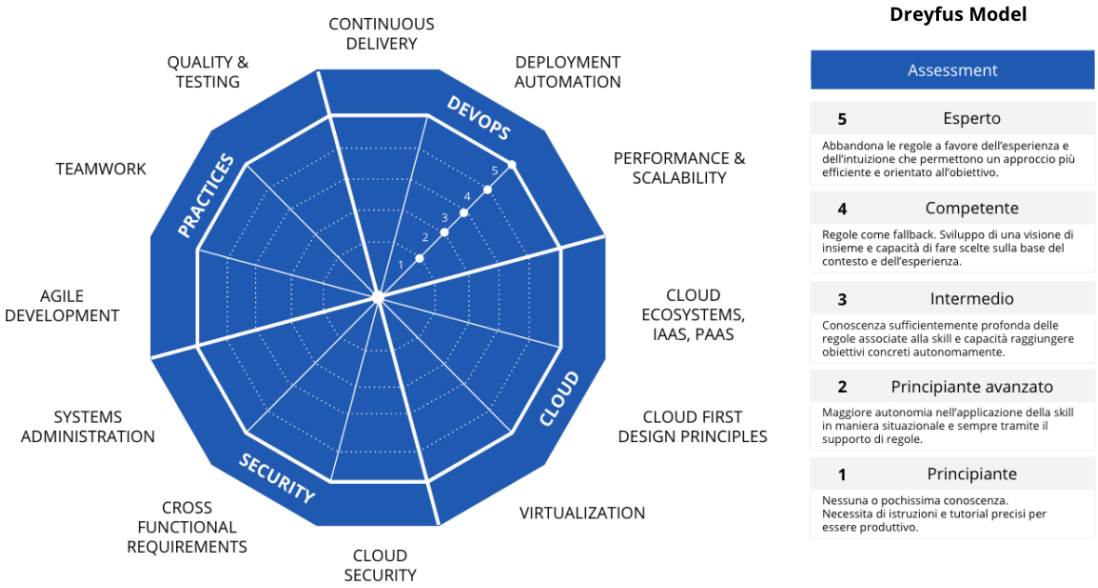
5. **esperto**: se si fa affidamento sull'esperienza ancor più che sulle regole e si ha dunque un approccio maturo, efficiente ed orientato all'obiettivo

Per avere una visione d'insieme e immediata delle competenze con i diversi livelli di maturità, si può poi costruire un diagramma di Kiviat, vedi a pagina 9 del workshop Mappatura Competenze e Indicatori di Performance in formato PDF o in formato ODP o in formato PPTX (conosciuto anche come grafico ragnatela o spider chart). Con un grafico di questo tipo si riescono a mostrare in un'unica rappresentazione:

- **le aree di competenza**: marcate sul grafico come parti della circonferenza, ovvero insiemi di raggi
- **le diverse skill**: i raggi del grafico, che hanno tutti origine dallo stesso centro
- **il livello di competenza di ciascuna skill**: marcato con un punto sul raggio proporzionale al livello della skill rispetto al massimo livello (ovvero 5 = esperto) raggiungibile
- **la situazione generale in cui l'amministrazione si trova in termini di skill possedute rispetto a skill carenti o assenti**: congiungendo i punti sui raggi con segmenti si ottiene infatti il famoso grafico a forma di ragnatela utile per una valutazione immediata della situazione attuale

Un esempio di questo tipo di grafico è presentato nell'immagine sotto, dove:

- devops, practices, security e cloud sono le aree di competenza
- continuous delivery, deployment automation, ecc. sono le diverse skill
- con i numeri da 1 a 5 sono i diversi livelli che possono essere marcati



5.2.3 4.2.3 Strumento per la valutazione delle competenze

Come anticipato sopra, proponiamo qui uno strumento che presenta un set di competenze necessarie già definite che possono essere usate dall'amministrazione come punto di partenza per la valutazione della propria situazione in termini di competenze per il cloud.

Consigliamo alle amministrazioni di utilizzare il file creato (vedi allegato "Valutazione competenze in formato ODS" o "Valutazione competenze in formato XLXS") per iniziare il proprio assessment che potrà essere eseguito sia in maniera autonoma da ogni singolo componente del team, evidenziando così le peculiarità di ognuno, sia valutando le competenze come team.

Il documento è suddiviso in diversi fogli:

- **Modifica skills:** foglio master per gli altri fogli del documento e che contiene tutti i nomi delle skills sulle quali verrà fatto l'assessment assieme alle loro descrizioni ed una valutazione del livello minimo per ogni strategia di migrazione, espresso su una scala da 0 a 5 dove:
  - 0 = non necessaria
  - 1 = principiante
  - 2 = principiante avanzato
  - 3 = intermedio
  - 4 = competente
  - 5 = esperto
- **Assessment:** foglio dove il team può procedere alla valutazione delle proprie competenze sfruttando la colonna verde **Competenze** e selezionando il valore più appropriato tra quelli riportati
- **Valutazione Retire, Re-purchase, Re-host, Re-platform, Re-architect:** ognuno di questi fogli riporta un diagramma di Kiviat (o spider chart) che mette a confronto i valori delle competenze risultanti dall'assessment con quelli considerati minimi per la strategia di migrazione che si sta osservando. In questo modo sarà possibile visualizzare a colpo d'occhio le skills su cui si ha un gap di competenze e sulle quali è quindi necessario un supporto esterno o di formazione. Si noti che non è inclusa la strategia di Retain in quanto non richiede azioni (dunque competenze) specifiche.

5.3 4.3 SLA richiesti ai servizi qualificati

Gli SLA (service level agreement, ovvero accordi sul livello di un servizio) sono un elemento importante di qualsiasi contratto con i fornitori di servizi. Oltre a formalizzare le aspettative su tipo e qualità del servizio, uno SLA definisce le misure da adottare quando non vengono soddisfatti gli standard pattuiti. Infatti, uno SLA è un documento che descrive il livello di servizio offerto da un fornitore a un cliente, indicando le metriche con cui viene misurato il servizio e misure o sanzioni da mettere in atto se i livelli concordati non sono soddisfatti.

Per redigere uno SLA per la fornitura di servizi cloud, è importante tenere in considerazione diversi aspetti che possono essere adeguati ai modelli di fruizione SaaS, PaaS e IaaS.

I servizi qualificati da AgID dovranno rispondere a tutta una serie di requisiti e SLA come definito nella Circolare n.2 del 9 Aprile 2018[40] per i servizi IaaS e PaaS e nella Circolare n.3 del 9 Aprile 2018[41] per i servizi SaaS. Di seguito riportiamo una più generica lista di metriche da tenere in sempre in considerazione nella definizione degli SLA per i servizi in cloud:

- **metriche di uptime:** rappresenta l'ammontare di tempo in cui il servizio offerto sarà disponibile ed accessibile ai suoi utenti. Viene rappresentato da una percentuale di disponibilità, ad esempio:
  - 99.9% uptime rappresenta 8.77 ore di downtime in un anno
  - 99.99% uptime rappresenta 52.60 minuti di downtime in un ann.
  - 99.999% uptime rappresenta 5.26 minuti di downtime in un anno
  - 99.9999% uptime rappresenta 31.5 secondi di downtime in un anno

[40] https://trasparenza.agid.gov.it/moduli/downloadFile.php?file=oggetto_allegati/181151234430O__OCircolare+2-2018_Criteri+per+la+qualificazione+dei+Cloud+Service+Provider+per+la+PA.pdf

[41] https://trasparenza.agid.gov.it/moduli/downloadFile.php?file=oggetto_allegati/181151237210O__OCircolare+3-2018_Criteri+per+la+qualificazione+di+servizi+SaaS+per+il+Cloud+della+PA+%28002%29.pdf

- **capacità del servizio:** rappresenta la capacità in termini di carico o di numero di utenti massimi che possono accedere al servizio e le opzioni per poter espandere questo numero
- **monitoring delle performance:** la definizione delle modalità con cui il fornitore monitorerà e riporterà le prestazioni del sistema
- **sicurezza:** è importante definire metriche relative alla sicurezza del dato e alla regolamentazione del suo accesso, come ad esempio:
  - protezione del dato
  - livelli di crittografia
  - politiche di backup
  - permessi di accesso e consultazione del dato
  - conformità alle normative vigenti
  - modalità e tempi di comunicazione del provider in caso di data breach
- **cambi di infrastruttura:** anche se nel caso di provider cloud i cambiamenti infrastrutturali sono molto più frequenti ed eseguiti in maniera trasparente, può essere utile richiedere al fornitore di essere informati in caso di aggiornamenti o cambiamenti rilevanti, così da pianificare e avvisare gli utenti di possibili downtime
- **supporto:** il supporto è una parte fondamentale dei rapporti con i fornitori di servizi cloud e si può suddividere in due metriche fondamentali:
  - **tempo di risposta:** rappresenta il tempo necessario al fornitore per rispondere ad una richiesta di supporto del cliente. Questo tempo è può essere variabile in base alla criticità della richiesta
  - **tempo di risoluzione:** rappresenta il tempo necessario al fornitore per risolvere e chiudere una richiesta di supporto del cliente

Tramite l'utilizzo di questi accordi è quindi possibile definire i cosiddetti Service Level Objectives (SLO) associati alle diverse metriche considerate come nella seguente tabella d'esempio:

Metrica	**SLO**	**Periodo**
Uptime	99.95%	1 anno
Tempo di risposta del supporto	75% delle richieste verrà presa in carico in meno di un minuto 85% delle richieste verrà presa in carico entro due minuti 100% delle richieste verrà presa in carico entro tre minuti	3 mesi

Questi accordi permettono a fornitore e amministrazione di definire chiaramente le rispettive aspettative in termini di qualità del servizio e supporto.
Il fornitore dovrà produrre report frequenti e su richiesta dell'amministrazione, evidenziando i livelli relativi alle metriche concordate, così da fugare qualunque tipo di dubbi in caso di controversie.

5.4 4.4 Lock-in

Il lock-in è un fenomeno di natura tecnica ed economica tale per cui un'organizzazione non riesce a svincolarsi facilmente da una scelta tecnologica precedentemente effettuata. Tra i fattori che più spesso lo causano vi sono:

- costi di transizione molto alti, che impediscono di cambiare tecnologia o fornitore se non ad un costo (non solo monetario) molto elevato
- mancanza di informazioni esaustive sul sistema attuale, che impedisce ad un nuovo fornitore di subentrare in maniera efficiente

Esistono diverse tipologie di lock-in, ma anche diverse pratiche per prevenirlo o almeno ridurlo. Esserne a conoscenza ancor prima di iniziare una migrazione al cloud può aiutare la pubblica amministrazione a mitigare i rischi e i costi connessi a questo fenomeno.

5.4.1 4.4.1 Tipologie di lock-in nella pubblica amministrazione

Il caso di lock-in più comune nell'ambito della pubblica amministrazione è quello legato ai fornitori. Solitamente, infatti, le pubbliche amministrazioni instaurano con quest'ultimi rapporti solidi e di lungo periodo, dai quali difficilmente si svincolano per non ritrovarsi a gestire i problemi connessi al cambio di un fornitore ben consolidato. Questi problemi rappresentano i cosiddetti costi di transizione e possono essere molteplici e di diversa natura. Troviamo costi di natura:

- **burocratica**: come ad esempio la progettazione e lo svolgimento della procedura di gara per la selezione di un nuovo fornitore
- **tecnologica**: come ad esempio la migrazione dei dati creati fino a oggi dal vecchio al nuovo fornitore
- **operativa**: come ad esempio i costi da pagare per eseguire l'operazione di migrazione dei dati di cui sopra

Nell' ambito dei fornitori di soluzioni tecnologiche ci sono diverse tipologie di lock-in a cui prestare attenzione:

- **lock-in sul dato**: può verificarsi nel caso vengano utilizzati software che non permettono di esportare i dati generati nel corso del loro utilizzo secondo formati considerati standard o che lavorano con formati proprietari. Questi software tendono a generare un forte lock-in a causa degli alti costi in cui si incorre per migrare i dati sui nuovi sistemi
- **lock-in sulla conoscenza**: può verificarsi nel caso ci sia una scarsa documentazione del software e delle decisioni prese e/o una carenza (o assenza totale) di materiale di formazione. Queste mancanze comportano una bassa condivisione della conoscenza acquisita dal fornitore e aumentano dunque il lock-in sullo stesso. In questo caso, infatti, la pubblica amministrazione è completamente dipendente dal fornitore iniziale per l'ottenimento delle suddette informazioni e non è in grado di fornirle in maniera semplice ad un nuovo affidatario
- **lock-in contrattuale:** può verificarsi nel caso vi siano clausole contrattuali che rendono costoso e difficile il cambio di fornitore o software

5.4.2 4.4.2 Come mitigare il lock-in

Vi sono una serie di pratiche tecnologiche, metodologiche e contrattuali che, se attuate dalla pubblica amministrazione e dai suoi fornitori sin dall'inizio di un affidamento, permetteranno di ridurre e mitigare il lock-in insieme ai costi e ai rischi ad esso associati.

4.4.2.1 Pratiche tecnologiche

Come buone pratiche tecnologiche da adottare per mitigare il rischio di lock-in da fornitore consigliamo di:

- garantire l'interoperabilità mediante l'esposizione di API come definito anche dalle linee guida di AgID[42] e mediante lo scambio di dati in formati standard come JSON o XML
- evitare l'utilizzo di formati di dato proprietari, ma favorire invece i formati standard e open

[42] https://www.agid.gov.it/it/infrastrutture/sistema-pubblico-connettivita/il-nuovo-modello-interoperabilita

- avere la possibilità di esportare (e possibilmente importare) i dati in un formato standard che garantisca l'interoperabilità e la portabilità ad altri sistemi
- favorire l'utilizzo di strumenti e framework di sviluppo open e diffusi rispetto a soluzioni sviluppate internamente e di difficile manutenibilità per altri fornitori
- nel caso di servizi PaaS o IaaS, favorire l'utilizzo di servizi basati su tecnologie open source e valutare l'utilizzo di servizi unici del cloud service provider solo se il valore ottenuto giustifica il maggior costo di switch (ovvero di cambiamento)
- garantire la portabilità dell'applicativo tramite l'utilizzo di stack tecnologici indipendenti dalla piattaforma
- favorire l'indipendenza degli applicativi dalle loro dipendenze esterne e di piattaforma come ad esempio hardware, sistema operativo o servizi specifici dei cloud service provider (PaaS o IaaS) tramite l'utilizzo di strumenti di containerizzazione e architetture che separino il core applicativo della piattaforma
- favorire l'uso di tecnologie open source
- in caso di software sviluppato ad-hoc per l'amministrazione, assicurarsi di avere la proprietà e l'accesso al codice sorgente

4.4.2.2 Pratiche metodologiche

Le pratiche metodologiche da adottare per mitigare il rischio di lock-in con il fornitore sono principalmente orientate alla condivisione della conoscenza e possono essere suddivise in due aree: documentazione e metodologia di lavoro.

Documentazione

Spesso giudicata poco utile e non manutenuta nel tempo, la documentazione gioca tuttavia un ruolo molto importante sulla riduzione del lock-in. Questa, infatti, è uno degli strumenti fondamentali che la pubblica amministrazione ha per preservare internamente per successive consultazioni la conoscenza generata dal fornitore.

La chiave per una documentazione efficace sta nel concentrarsi solo sugli aspetti chiave e realmente importanti che la rendano così uno strumento strategico e non un ulteriore costo non necessario. Le informazioni importanti che devono essere documentate sono di vario tipo. Tra queste troviamo:

- **decisioni significative**: decisioni impattanti che generalmente possono essere riconosciute dai seguenti aspetti:
  - richiedono molto tempo per essere concordate
  - sono critiche nel raggiungimento di un requisito di progetto
  - il loro impatto nel medio-lungo termine non è chiaro da subito
  - impattano una grossa parte del sistema in oggetto
  - una volta prese ed attuate è difficile tornare alla situazione precedente

Per natura, queste decisioni si distinguono in:

- **architetturali**: decisioni di progettazione software che soddisfano un requisito funzionale o non funzionale e che hanno un impatto significativo sull'architettura del sistema
- **relative al business**: decisioni che rientrano nell'ambito più strategico dell'amministrazione o di un progetto specifico

I formati che consigliamo di adottare per questo tipo di decisioni sono l'**Architectural Decision Records** (ADR[43]) e il corrispondente **Business Decision Records** (BDR). Questi "record" nel tempo formeranno una memoria storica e cronologica delle decisioni prese, risulta quindi fondamentale l'immutabilità del contenuto e il fatto che non debbano mai essere cancellate, anche quando le decisioni sono state sostituite da altre. In allegato a queste linee guida trovate un possibile template per ADR/BDR (vedi allegato "ADR-BDR Template").

[43] https://adr.github.io

Questo tipo di documentazione può essere conservata direttamente all'interno dei repository di codice sorgente, se a scriverla sono persone coinvolte nello sviluppo software, altrimenti si consiglia di crearla e conservarla con un software specifico per la gestione della documentazione.

- **artefatti generati durante il processo di migrazione**: artefatti che devono essere conservati in aggiunta alle decisioni significative prese per tenere traccia dell'importante conoscenza generata durante il processo di migrazione. Alcuni esempi non esaustivi sono:
  - diagramma architetturale per visualizzare l'organizzazione delle risorse infrastrutturali on-premise e/o in cloud, evidenziando in particolare:
    - macchine virtuali
    - database
    - storage
    - networking
    - sistemi di bilanciamento del carico
    - sistemi di backup
    - servizi specifici del cloud service provider
  - procedure di alerting e monitoring
  - procedure di rilascio
  - procedure di patching
  - procedure di review della sicurezza
  - guida alla gestione delle identità e rispettivo accesso ai sistemi (IAM)
  - documentazione delle API dei servizi
  - documentazione delle personalizzazioni sviluppate sugli applicativi
- **materiale di formazione**: materiale informativo sui software adottati dall'amministrazione, per evitare che il fornitore sia il detentore unico di questa conoscenza. La presenza di documentazione che evidenzi come interagire ed eventualmente amministrare il software permetterà all'amministrazione di essere più autonoma e veloce nell'organizzazione e formazione dei suoi impiegati, avvalendosi del supporto del fornitore solo in casi particolari e più complessi.

Uno dei principali rischi associato alla documentazione è che essa smetta di essere manutenuta regolarmente diventando, nel peggiore dei casi, una fonte di informazioni obsolete e soprattutto sbagliate. Una tecnica utile a mitigare questo rischio è quella di creare meeting ricorrenti in cui fare revisione delle informazioni documentate fino a quel momento, così da tenere sempre alto il livello di attenzione su questo strumento strategico e identificare tempestivamente eventuali mancanze.

Metodologia di lavoro

Lavorare insieme ai fornitori seguendo buone pratiche di collaborazione e condivisione può ridurre significativamente il rischio di lock-in. Per questo motivo, consigliamo di adottare le seguenti buone pratiche:

- **incontri di allineamento**: incontri ricorrenti che aiutino a raggiungere un buon grado di condivisione della conoscenza riducendo i rischi connessi al lock-in su di essa. In particolare, consigliamo di organizzare i seguenti incontri di allineamento:
  - **showcase:** un incontro di circa un'ora da organizzare alla fine di ogni iterazione (ogni una o due settimane) durante il quale si presentano i risultati raggiunti e il valore prodotto nell'iterazione corrente e si discute il piano per l'iterazione successiva integrando eventuali feedback (vedi capitolo 5.1.4)

    Partecipanti: team del fornitore, team tecnico dell'amministrazione, responsabili di progetto dell'amministrazione

Durata massima: 60 minuti

- **tech review:** meeting in cui si riuniscono il personale tecnico dell'amministrazione e del fornitore per poter condividere considerazioni importanti in merito ad architettura e codice del software o decisioni importanti dal punto di vista infrastrutturale.

  Partecipanti: team tecnico del fornitore, team tecnico dell'amministrazione

  Durata massima: 60 minuti

- **pairing:** tecnica in cui due persone lavorano insieme per risolvere un problema così da condividere opinioni e soluzioni in maniera trasparente e rapida. Nello specifico ambito dello sviluppo software questa tecnica viene denominata "Pair Programming". Uno dei suoi vantaggi è la facilità con cui si condividono conoscenze e contesto aumentando nel complesso il flusso dell'informazione all'interno del team e aiutando a conservare la conoscenza all'interno dell'amministrazione
- **visual management:** tecnica di comunicazione che attraverso l'utilizzo di artefatti visuali punta a comunicare informazioni chiave per il team. Nell'ambito di una migrazione può essere usata, ad esempio, per la visualizzazione dei lavori in corso con una Kanban board (vedi capitolo 5.1.4). L'aspetto chiave di questa tecnica è appunto l'utilizzo di informazioni visuali come rimedio all'ambiguità interpretativa delle comunicazioni verbali.

4.4.2.3 Pratiche contrattuali

I criteri per la qualificazione dei Cloud Service Provider per la PA[44], richiedono esplicitamente ai CSP di garantire l'assenza di ogni tipo lock-in dell'Acquirente nei confronti del Fornitore Cloud. È bene però integrare questo primo livello di mitigazione del rischio considerando le pratiche tecnologiche e metodologiche illustrate sopra ancor prima di iniziare l'affidamento.

In particolare, nella fase di pianificazione dell'affidamento, è necessario che l'amministrazione non solo definisca i bisogni funzionali relativi al software richiesto, ma consideri anche, oltre al costo iniziale, i costi futuri relativi alla manutenzione della soluzione e alla sua potenziale migrazione su nuovi sistemi o su fornitori diversi.

Si consiglia quindi, con lo scopo di mitigare i rischi e i costi connessi al lock-in, di introdurre nei documenti di gara le opportune clausole relative alle buone pratiche presentate in questo capitolo.

[44] https://trasparenza.agid.gov.it/moduli/downloadFile.php?file=oggetto_allegati/181151234430O__OCircolare+2-2018_Criteri+per+la+qualificazione+dei+Cloud+Service+Provider+per+la+PA.pdf

CAPITOLO 6

5. Eseguire la migrazione: gli applicativi

La migrazione al cloud è una sfida che riguarda aspetti tecnologici, di processo ed in particolar modo culturali, e il successo dell'operazione è legato al superamento degli ostacoli in ognuno di questi ambiti.

Nel capitolo 4 abbiamo illustrato le modalità con cui è possibile affrontare questa sfida da un punto di vista tecnologico (ovvero le possibili strategie di migrazione) e gli elementi chiave da considerare nella fase di pianificazione di una migrazione.

In questo capitolo, invece, entriamo nella parte operativa della migrazione al cloud e andiamo ad approfondire i temi legati alla sua esecuzione. Per poterla eseguire nel migliore dei modi, infatti, è importante considerare aspetti che aiutano a ridurre il rischio di fallimento e aumentano il valore prodotto con il medesimo sforzo.

Prima di tutto, trattiamo qui il metodo di lavoro. Questo, infatti, deve fare propri dei concetti mirati a mantenere allineati il più possibile gli obiettivi di chi esegue la migrazione, di chi usufruirà del sistema migrato e di chi supporta l'iniziativa, a condividere con tempestività rischi e problematiche che emergono strada facendo in modo da facilitare decisioni consapevoli e ridurre il rischio di dispendio di energie in direzioni errate, e a migliorare il processo di lavoro a fronte delle informazioni apprese durante l'attività e gli insegnamenti provenienti dall'esperienza diretta.

Prima di procedere con la migrazione effettiva è importante avere ben chiari gli obiettivi che si vogliono raggiungere per poter focalizzare le proprie energie sulle soluzioni possibili per poterli ottenere, misurare lo stato da cui si parte e rilevare poi gli stessi indicatori in modo continuativo dopo la migrazione per valutare i progressi ottenuti.

Le soluzioni tecnologiche, invece, devono tenere in considerazione le differenti sfide che un ambiente in cloud pone rispetto ad uno on-premise e mirare allo sfruttamento ottimale delle potenzialità del primo. Per questo motivo, presentiamo di seguito le buone pratiche da seguire per gestire gli aspetti rilevanti del sistema. Per quanto riguarda uno degli aspetti più critici, ovvero la migrazione dei dati, rimandiamo invece al capitolo 6 dove abbiamo approfondito il tema con esempi e linee guide specifiche per i diversi scenari che la pubblica amministrazione può trovarsi ad affrontare.

Infine, per considerare una migrazione effettivamente completata non si può prescindere dalla validazione che tutto ciò che è stato fatto fornisca effettivamente agli utenti lo strumento di cui hanno bisogno, che l'utilizzo della piattaforma in cloud funzioni secondo le modalità attese o, auspicatamente, anche meglio e che l'erogazione dei servizi forniti dagli applicativi migrati siano in linea con gli obiettivi che si erano identificati. L'ultima sezione di questo capitolo è dedicata ai metodi di validazione e ai metodi che permettono di verificare i cambiamenti attuati così da ridurre i rischi e garantire il corretto funzionamento del sistema.

6.1 5.1 Metodologia di lavoro

La migrazione al cloud è una sfida di tipo tecnologico ma, in modo rilevante, anche culturale e di processo.

La metodologia di lavoro, ovvero l'insieme di quelle pratiche con cui il team approccia e gestisce l'analisi, la progettazione e l'effettiva migrazione al cloud degli applicativi, rientra a pieno titolo tra i fattori da considerare a livello culturale e di processo.

6.1.1 5.1.1 Team cross-funzionale

Il team di progetto deve includere tutte le competenze necessarie per poter eseguire le attività relative alla migrazione e trovare le soluzioni necessarie durante l'esecuzione.

Questo team cross-funzionale deve comprendere non solo competenze tecniche ma anche competenze normative, di processo, di comunicazione o, in senso più generale, necessarie ad affrontare le principali problematiche che si possono manifestare nell'esecuzione dell'iniziativa.

6.1.2 5.1.2 Iteratività e incrementalità

Tutte le fasi di un piano di migrazione devono essere *iterative* ed *incrementali* rispetto ad un insieme *prioritizzato* di azioni che si vogliono intraprendere sia nel caso si stia definendo il piano di migrazione degli applicativi, sia nel caso si stia eseguendo una specifica migrazione.

Nel caso si stia definendo il piano di migrazione è opportuno:

1. prioritizzare gli applicativi secondo i criteri suggeriti in questo documento
2. identificare la strategia di migrazione per l'applicativo a priorità più alta
3. analizzare i gap di competenze necessarie per eseguire la migrazione
4. se la fattibilità tecnica, operativa e di competenze è confermata, procedere con l'esecuzione della migrazione oppure passare all'analisi dell'applicativo con priorità successiva
5. nel caso si sia eseguita la migrazione, alla luce dei risultati ottenuti, rivedere la prioritizzazione effettuata inizialmente
6. con il passare del tempo possono altresì mutare delle condizioni che hanno definito la prioritizzazione o l'analisi di dettaglio di un applicativo (es. valore generato da un applicativo, identificazione di applicativi non ancora censiti, ...). Per questo motivo è consigliabile rivedere con cadenza regolare l'elenco degli applicativi da migrare e ripartire con il processo dall'inizio

Nel caso si stia eseguendo una specifica migrazione è opportuno che:

1. le iterazioni siano di poche settimane (una o due)
2. al termine di ogni iterazione, si mostri il progresso attraverso il rilascio di piccole modifiche incrementali che mantengano l'applicativo funzionante
3. il risultato dell'iterazione sia validato con gli stakeholder del progetto per ottenere un riscontro con cadenza regolare sull'avanzamento dell'attività. Ciò consente di intercettare problemi o opportunità di miglioramento con tempestività riducendo significativamente il costo di correzione del problema e permettendo di riprioritizzare le azioni da intraprendere sulla base dell'opportunità identificata
4. l'elenco delle attività sia riprioritizzato rispetto ai nuovi problemi da fissare e/o alle nuove opportunità emerse

6.1.3 5.1.3 DevOps

Sviluppatori, sistemisti e tester devono adottare un approccio operativo conforme alle pratiche DevOps[45] per garantire comunicazione, collaborazione e integrazione tra sviluppatori e addetti alle operations.

DevOps risponde all'interdipendenza tra sviluppo software e IT operations, ed aiuta un'organizzazione a sviluppare in modo più rapido ed efficiente prodotti e servizi software.

Un'amministrazione pubblica può trovarsi in uno di questi scenari per un applicativo:

1. controlla le attività di sviluppo (dev) e di deployment/messa in produzione (ops)
2. controlla le attività di deployment/messa in produzione (ops) e delega ad un soggetto terzo lo sviluppo (dev)
3. delega ad un soggetto terzo sia le attività di sviluppo che di deployment
4. acquista una soluzione di mercato e ne gestisce il deployment sulla propria infrastruttura
5. acquista una soluzione erogata da terze parti

Nei primi tre scenari si raccomanda di prevedere pratiche DevOps a livello organizzativo e/o contrattuale. Nello scenario 4. si suggerisce di richiedere evidenza delle pratiche adottate dal produttore prima dell'acquisto e di lavorare congiuntamente per la realizzazione di un processo integrato di deployment. Nello scenario 5. si raccomanda di richiedere evidenza delle pratiche adottate dal produttore prima dell'acquisto.

6.1.4 5.1.4 Collaborazione e confronto continuo

Le parti interessate, o stakeholders, ed il team di progetto devono congiuntamente definire:

- gli **obiettivi**
- gli **indicatori** per la loro misurazione
- il **valore** che si vuole generare per gli utenti finali

all'inizio dell'iniziativa di migrazione e mantenerli aggiornati durante l'esecuzione a fronte delle criticità ed i punti di attenzione che via via emergeranno.

Le parti interessate devono altresì validare i risultati presentati al termine di ogni iterazione dal team di progetto in modo da identificare il prima possibile elementi che richiedono correzioni o variazioni e discutere le criticità emerse che condizionano il raggiungimento dei risultati attesi, fino a ridefinire gli obiettivi o il valore atteso se necessario.

Figure rilevanti da coinvolgere nel processo sono i rappresentanti degli utenti finali che beneficeranno degli effetti della migrazione, sia che questa impatti i processi sia che questa riguardi solo i dati. Il loro coinvolgimento favorisce la validazione dei risultati, la comprensione, l'accettazione e la diffusione fra gli altri utenti del processo di cambiamento in atto.

È importante sottolineare che l'aspetto cruciale è l'interazione fra gli individui piuttosto che i processi e gli strumenti usati.

Una volta avviato il progetto, il team (probabilmente composto sia da dipendendenti della pubblica amministrazione che da fornitori) dovrà gestirsi e coordinarsi autonomamente.

Per facilitare la collaborazione, le seguenti pratiche dovrebbero essere adottate e adattate a seconda del contesto:

- **visualizzazione del lavoro in corso:** tramite una Kanban board fisica o digitale che permetta di avere visione complessiva del lavoro programmato, in corso e concluso per l'iterazione corrente
- **stand-up giornaliero**: un breve meeting di 5-15 minuti (a seconda della dimensione del team) da organizzare al mattino durante il quale ciascun membro del team dà un breve aggiornamento sui progressi ed eventuali blocchi del giorno precedente e sintetizza gli obiettivi per la sua giornata

[45] https://cloud.italia.it/glossario/#devops

- **showcase alla fine di ogni iterazione**: un incontro di circa un'ora da organizzare alla fine di ogni iterazione (ogni una o due settimane) durante il quale si presentano i risultati raggiunti e il valore prodotto nell'iterazione corrente e si discute il piano per l'iterazione successiva integrando eventuali feedback.
- **retrospettiva**: un altro meeting ricorrente di circa un'ora da organizzare alla fine di ogni iterazione durante il quale il team ha l'occasione di analizzare, in retrospettiva appunto, l'iterazione appena conclusa con l'obiettivo di migliorare continuamente sia il processo che l'esecuzione, tenendo in considerazione i feedback di ciascun membro del team

6.1.5 5.1.5 Miglioramento continuo

La pratica della retrospettiva è stata introdotta per rispondere alla necessità di effettuare riflessioni regolari sull'andamento del progetto, in modo da correggerne il funzionamento prima che sia finito. Questo in contrasto con le tradizionali review post-mortem, che danno informazioni utilizzabili solo in progetti seguenti e che non aiutano dunque a raggiungere il successo nei progetti in corso.

La retrospettiva è conosciuta come pratica della metodologia Agile in quanto uno dei principi del manifesto Agile[46] è incentrato proprio sulla riflessione in team: «Ad intervalli regolari il team riflette su come diventare più efficace, dopodichè regola e adatta il proprio comportamento di conseguenza.»

Lo scopo generale di una retrospettiva è stimolare l'analisi e la riflessione e incoraggiare il miglioramento continuo. In particolare, gli obiettivi sono:

- aumentare il livello di interazione e condivisione del team
- identificare cosa è andato bene e cosa non ha funzionato in relazione a processi, strumenti e dinamiche di gruppo
- discutere e scoprire opportunità di miglioramento e definire un piano d'azione per implementarlo
- enfatizzare e dare uguale tempo di valutazione e condivisione a ciò che andato bene. È importante concentrarsi sul positivo e identificare ciò che è andato bene in modo da continuare a farlo

La retrospettiva è un modo per spronare i membri del team a riflettere su quello che succede intorno a loro nel corso del progetto e ad identificare azioni che possano migliorare sia il processo che l'esecuzione. In concreto, si tratta di un incontro di circa un'ora che si tiene alla fine di una iterazione durante il quale il team riflette su quanto accaduto durante l'iterazione appena trascorsa e individua le azioni per migliorare la successiva. Si consiglia di eseguire la retrospettiva con regolarità includendo sia il team tecnico dell'amministrazione che il team tecnico del fornitore.

Per eseguire una retrospettiva efficace è importante trovare la tecnica ed il formato più adatto al contesto. Per approfondire puoi consultare il libro di Esther Derby "Agile Retrospectives: Making Good Teams Great" o altre risorse disponibili online, come ad esempio questo articolo[47].

6.2 5.2 La preparazione

Prima di iniziare la migrazione vera e propria di un applicativo e dei suoi dati verso una piattaforma in cloud, è necessario considerare una serie di attività preparatorie, in modo da ridurre il rischio di problemi nella transizione che porterebbero ad un conseguente aumento dei tempi per poterla portare a termine.

Le attività preparatorie mirano ad identificare eventuali problematiche da analizzare e risolvere prima di proseguire. La mancata risoluzione di queste potrebbe essere anche motivo per valutare una differente strategia di migrazione.

Quando si vuole migrare un applicativo in cloud è importante seguire le seguenti best practice:

[46] https://agilemanifesto.org/
[47] http://www.marioconcina.it/blog/come-fare/5-tecniche-di-retrospettive-agile.html

1. pianificare l'ottimizzazione dei costi identificando chiaramente le aree ed i componenti su cui è possibile ottenere un vantaggio in termini di costo una volta in cloud rispetto alla situazione corrente (ad es. stimare accuratamente la dimensione dei dati e la loro rapidità di incremento, analizzare le modalità di costo per utilizzo effettivo, risparmio energetico, riduzione investimenti per hardware e networking, tempo risparmiato per la gestione dei backup e disaster recovery, . . .) stimando la spesa attuale ed il risparmio atteso. Questo permetterà di misurare, a migrazione completata, i nuovi valori e confrontarli con i precedenti per valutare l'efficacia dell'operazione
2. identificare le interdipendenze dell'applicativo per avere una chiara comprensione di come gli applicativi e l'infrastruttura collaborino tra di loro. Questo è da guida nel processo di migrazione, per identificare quali applicazioni migrare ed in quale ordine
3. verificare che l'applicativo sia compatibile con l'eventuale nuova versione del sistema operativo
4. correggere i bug ancora aperti
5. definire il test plan per verificare, a migrazione terminata, che:
   a. l'applicativo funzioni correttamente prima di migrare i dati
   b. la nuova versione utilizzi correttamente eventuali nuovi componenti introdotti per sostituire parti dell'applicativo on-premise
   c. l'applicativo funzioni correttamente con i dati migrati
6. Fare l'assessment della base dati: la maggior parte dei cloud provider forniscono un tool di assessment per la base dati che supporta la creazione di un report per la loro migrazione. Il report contiene valutazioni sulle licenze, le funzionalità supportate in cloud, la configurazione dell'hardware e le azioni intraprese per la migrazione. Questo report permette di migliorare performance e disponibilità della base dati
7. assicurarsi che il cloud provider fornisca meccanismi di import ed export di tutti i dati.

6.3 5.3 Buone pratiche

6.3.1 5.3.1 Scalabilità

La scalabilità è la capacità di un applicativo di gestire i carichi di operatività: è la possibilità di adattare le risorse, aumentandole o diminuendole, in base al bisogno, on demand.

Le piattaforme cloud, in base al livello di integrazione di un applicativo, forniscono un'esperienza di gestione più semplice, rapida e mirata sia per incrementare le risorse quando il traffico lo richiede che per ridurle quando la necessità non è più presente mentre in caso di applicativi SaaS la gestione della scalabilità è demandata al fornitore del servizio che se ne occupa in completa autonomia.

Applicativi in cloud si adattano meglio all'evoluzione dei bisogni del servizio finale.

La scalabilità di un applicativo può essere:

- orizzontale, ovvero la capacità di supportare maggiore traffico aggiungendo ulteriori macchine all'insieme già attivo.

  A seconda del livello di servizio offerto dal cloud provider sarà ad esempio possibile aggiungere e rimuovere macchine virtuali o container applicativi senza doversi occupare direttamente dell'infrastruttura sottostante, oppure, tramite i più avanzati servizi di autoscaling, fornire delle policy specifiche sulla base delle quali il CSP provvederà ad aumentare o ridurre il numero di macchine o container necessari così da garantire il livello di servizio desiderato.
- verticale, ovvero la capacità di supportare maggiore traffico aggiungendo più risorse alle macchine già attive. La scalabilità verticale non rende un sistema fault tolerant: applicativi che funzionano utilizzando una singola macchina smettono di funzionare se quella macchina ha un down time. Anche in questo caso a seconda del livello di servizio offerto dal cloud provider sarà ad esempio possibile aumentare o ridurre risorse come CPU,

memoria o storage riavviando gli applicativi o nei servizi più avanzati applicando una modifica a caldo, cioè senza dover riavviare.

6.3.2 5.3.2 Disponibilità

La disponibilità di un applicativo è la sua capacità di essere funzionante nel momento in cui vi è necessità di utilizzo. La disponibilità si misura in uptime e le piattaforme cloud offrono meccanismi innovativi per ottenere disponibilità molto elevate, prossime al 100%.

La disponibilità di un applicativo è ottenuta con:

- il deploy di più istanze per ogni servizio:
  - i componenti dell'applicativo, meccanismo di autenticazione compreso, devono essere deployati su più istanze per evitare un unico punto di vulnerabilità, *single point of failure*
  - almeno un'istanza per ogni componente (load balancer, application server, database) deve essere presente in *due zone differenti*
  - se possibile, avere una capacità garantita in regioni separate, cioè aree territoriali formate da data center indipendenti tra loro, spesso suddivise a loro volta in zone di disponibilità fisicamente separate ma collegate tra loro da connessioni ad alta affidabilità, prestazioni e ridondanza;
  - i dati devono essere replicati tra le region, se necessario, per avere un meccanismo di *failover*, ovvero la sostituzione della region non più funzionante con quella funzionante in caso di guasto o interruzione anomala
- il deploy dell'applicativo su più region per evitare che, nel caso di un applicativo deployato in una sola region, se questa diventa indisponibile anche l'applicativo è indisponibile, impattando l'uptime degli SLA
- testando ed automatizzando il deploy utilizzando tool e scripts che aggiornano e validano la configurazione ed automatizzano il deployment. Anche gli aggiornamenti devono essere realizzati in modo automatizzato. Assicurarsi di aver rafforzato e ristretto le politiche di deployment al fine di minimizzare i cambiamenti manuali apportanti da operatori.

6.3.3 5.3.3 Resilienza

La resilienza è la capacità di gestire i malfunzionamenti limitandone l'impatto e gestendo il degrado in modo graduale e ripristinare successivamente il corretto funzionamento del sistema.

Per ottenere servizi resilienti è cruciale:

- Identificare i malfunzionamenti e ripristinare il corretto funzionamento in modo rapido ed efficace
- isolare i componenti in modo che il malfunzionamento di uno non impatti gli altri
- suddividere i servizi in gruppi, includendo in ogni gruppo tutti quelli necessari, ed allocare le risorse separatamente per ogni gruppo in modo che un malfunzionamento non impatti i servizi esterni allo specifico gruppo
- includere i servizi necessari in un gruppo sulla base di requisiti tecnici o funzionali

La resilienza di un applicativo in cloud è superiore grazie a:

- ridondanza
- autoscaling, ovvero abilitando la disponibilità di un applicativo in più zone, l'autoscaling aiuta a dimensionare la capacità sulla base della richiesta effettiva:
  - scaling policies (CPU, memoria)

  - scaling programmato
- monitoring per verificare che il comportamento corrisponda a quello atteso
- replication, ovvero la possibilità di replicare i servizi importante per assicurarsi che sia disponibili in qualsiasi momento

La resilienza di un applicativo è ottenuta seguendo queste best practice:

- analizzare continuamente il sistema per identificare i malfunzionamenti, il loro impatto e le modalità di ripristino del funzionamento atteso
- utilizzo di load balancer per distribuire il carico
- utilizzo di più zone per la disponibilità
- monitoraggio dello stato di salute delle dipendenze e degli endpoint
- progettare il proprio applicativo secondo i principi del design for failure
- preparare la documentazione per il failover ed il fallback, ovvero una soluzione alternativa nel caso quella principale non sia disponibile

6.3.4 5.3.4 Sicurezza

Le piattaforme cloud, diversamente dalle soluzioni on-premise, sono intrinsecamente caratterizzate dalla condivisione di risorse, ponendo quindi maggiore criticità agli aspetti di sicurezza.

Problematiche come data leakage (ovvero trasmissione non autorizzata di dati dall'interno di un applicativo ad un destinatario esterno), un controllo debole degli accessi, attacchi DDoS, data breaches (ovvero dati sensibili, protetti o riservati vengono consultati, copiati, trasmessi, rubati o utilizzati da un soggetto non autorizzato), la perdita di dati per errori o negligenza, la gestione delle identità e della privacy devono essere tenute in forte considerazione.

Per mitigare questi rischi, le piattaforme cloud forniscono un insieme di policy, controlli e regole che assieme proteggono l'infrastruttura con misure specificatamente destinate alla sicurezza.

La sicurezza di un applicativo può essere migliorata seguendo queste best practice:

- mettere in sicurezza tutte le risorse, non solo quelle esposte verso l'esterno, *edge layer* (es. utilizzando una connessione TLS sicura anche nelle comunicazioni con altri applicativi)
- proteggere i dati memorizzati, *data in rest*, in qualsiasi forma digitale (es. database, data warehouse, spreadsheet, archivi, nastri, backup, dispositivi mobile, ecc.) attraverso la criptazione
- mitigare attacchi DDoS utilizzando il livello di network della piattaforma cloud;
- utilizzare una lista di accessi sicuri per reti, applicativi e dati
- eseguire un'analisi periodica delle vulnerabilità e i penetration test
- utilizzare *two factor authentication* (2fa) e configurare un meccanismo di *single sign on* (SSO)
- installare antivirus e anti-malware per i nodi e il networking
- abilitare il monitoring ed il logging per il networking, gli applicativi ed i dati
- connettere on-premise con cloud utilizzando sempre un link dedicato ed una VPN sul link pubblico

5.3.4.1 Sicurezza del dato

Con la conservazione dei dati in cloud, è importante valutare il livello di protezione dei propri applicativi e quali modifiche o controlli debbano essere implementati per poter operare in sicurezza. A riguardo, si consiglia di mettere in pratica le seguenti best practices:

- criptazione dei dati prima di passare al cloud
- criptazione dei dati memorizzati nei dischi utilizzando AES (Advanced Encryption Standards) 256
- utilizzo del Key-Manager fornito dalla piattaforma cloud per la memorizzazione dei dati sensibili come credenziali, token per le API, certificati SSL, chiavi private
- controllare gli accessi sulla base del ruolo degli utenti
- proteggere tutti i canali di comunicazione con un certificato SSL

6.3.5 5.3.5 Data Privacy

I dati sono un aspetto molto critico per un'organizzazione e conservarli in cloud impone l'adozione di misure per mantenerli sicuri e sotto il proprio controllo visto che vengono effettivamente memorizzati in modo distribuito su diverse macchine e sistemi di storage.

La data privacy viene garantita tramite questi aspetti:

- data integrity, impedendo che persone o applicativi non autorizzati possano modificare o cancellare i dati. Questo può essere ottenuto attraverso l'implementazione di meccanismi di controllo degli accessi, sistemi di controllo delle versioni dei dati che impediscano la perdita del dato originale a seguito di modifiche o cancellazione, l'utilizzo di *checksum* per verificare l'integrità;
- data confidentiality, proteggendo i dati contro l'accesso non autorizzato o il furto. Questo può essere ottenuto attraverso l'implementazione di meccanismi di controllo degli accessi e criptazione dei dati;
- data availability, assicurando la disponibilità e l'accessibilità dei dati quando è necessario. Questo può essere ottenuto attraverso le misure disponibili in cloud come SLA per alta affidabilità, ridondanza e business continuity.

6.3.6 5.3.6 Autenticazione ed autorizzazione

L'autenticazione, l'autorizzazione e l'auditing in cloud permettono di avere il controllo dell'applicativo quando questo è deployato in cloud.

L'autenticazione è il processo di conferma dell'identità di un utente in cui l'applicativo determina chi sta accedendo in base all'utilizzo di credenziali valide. Tipicamente l'autenticazione è fatta utilizzando username e password, ma esistono altri metodi di autenticazione:

1. *Single Factor*: è il metodo di autenticazione più elementare, comunemente utilizzato per accedere un sistema come un sito web attraverso delle credenziali (es. username e password)
2. *Multi Factor:* è un metodo di autenticazione che fornisce l'accesso ad un utente solo dopo aver presentato con successo due o più dimostrazioni della propria identità sulla base di: conoscenza (qualcosa che l'utente e solo lo specifico utente sa), possesso (qualcosa che l'utente e solo lo specifico utente ha) e inerenza (qualcosa che l'utente e solo lo specifico utente è)
3. *Two Factor* (2FA)*:* conosciuto anche come verifica in due passaggi o autenticazione con doppio fattore, è un tipo, o un sottoinsieme di multi-factor authentication. È un modo di confermare l'identità dichiarata dagli utenti utilizzando la combinazione di due diversi fattori fra:
    a. qualcosa che loro conoscono
    b. qualcosa che loro posseggono
    c. qualcosa che sono

L'autenticazione con due fattori aggiunge un livello di sicurezza agli applicativi rendendo più difficile ottenerne l'accesso per chi prova ad attaccarli.

6.3.7 5.3.7 Interoperabilità

Durante un processo di migrazione si deve considerare l'opportunità di adeguare l'applicativo al "Nuovo modello di interoperabilità[48]" per renderlo più facilmente integrabile con gli altri enti.

Nel caso di strategia di migrazione re-purchase, ovvero nel caso di acquisto di servizi SaaS, l'interoperabilità è un criterio che deve essere preso in considerazione: se il SaaS acquisito si interfaccia direttamente con altre amministrazioni, deve farlo in conformità alle Linee guida sull'interoperabilità tecnica,[49] alternativamente l'ente dovrà sviluppare sopra il SaaS un livello di interoperabilità.

Tutti i servizi presenti sul cloud Marketplace[50] di ACN, la piattaforma che espone i servizi e le infrastrutture qualificate, sono compatibili con altri servizi e infrastrutture cloud dello stesso tipo mediante l'utilizzo di standard aperti (ad esempio Open Virtualization Format) ed opportune API.

Nel caso di strategia di migrazione re-platform, re-architect e/o per lo sviluppo di nuovi applicativi cloud-native, invece, le Pubbliche Amministrazioni devono seguire le Linee guida sull'interoperabilità tecnica.[51] che garantisce la collaborazione tra le Pubbliche Amministrazioni e verso soggetti terzi.

6.3.8 5.3.8 Monitoraggio e alerting

E' essenziale che l'intera infrastruttura funzioni in modo efficiente e che le risorse del cloud vengano utilizzate in modo efficace. Per questo motivo, esistono tecniche di misurazione e controllo che permettono di monitorare la stabilità e le prestazioni di un'infrastruttura.

Le metriche più usate si basano sia su dati degli applicativi (come attività degli utenti) che su dati del sistema (come registro degli eventi).

Soprattutto in ottica di infrastruttura e applicativi cloud, i principali dati da monitorare sono:

- Azioni fallite e riuscite eseguite dagli utenti
- Disponibilità dell'applicativo
- Performance dell'applicativo
- Integrità dei file
- Tentativi falliti e riusciti di accesso ai dati e alle risorse
- Attività sospette e illecite
- Informazioni di base sull'infrastruttura (CPU, RAM, disco, network, performance)
- Costi (budget) del cloud provider scelto

Tra le varie metriche basate sui dati di cui sopra, ce ne sono tre fondamentali da considerare per misurare la disponibilità, l'affidabilità e la resilienza di un applicativo e valutare i rischi connessi:

- Tempo medio al guasto (conosciuto come MTTF, mean time to failure), che misura il tempo medio del verificarsi di un guasto o malfunzionamento del sistema, ovvero il tempo medio di uptime
- Tempo medio tra i guasti (conosciuto come MTBF, mean time between failures), che misura il tempo medio di attesa tra un guasto e il successivo
- Tempo medio di ripristino (conosciuto come MTTR, mean time to repair), che misura il tempo necessario a riparare un componente o una parte guasta del sistema

Per la scelta di un sistema di monitoraggio, si consiglia di:

[48] https://www.agid.gov.it/it/infrastrutture/sistema-pubblico-connettivita/il-nuovo-modello-interoperabilita
[49] https://docs.italia.it/italia/piano-triennale-ict/lg-modellointeroperabilita-docs
[50] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud
[51] https://docs.italia.it/italia/piano-triennale-ict/lg-modellointeroperabilita-docs

1. Analizzare l'infrastruttura e definire i requisiti di monitoraggio per il proprio ambiente e applicativo
2. Allocare un budget specifico per il monitoring e comparare i costi delle soluzioni offerte sul mercato per trovare la soluzione che soddisfa i requisiti di funzionalità e di budget
3. Eseguire un test pilota del sistema di monitoraggio su un applicativo per verificare le funzionalità in uno scenario reale

In generale, un sistema di monitoraggio deve essere semplice da usare e fornire una chiara visualizzazione delle informazioni, che devono essere rese immediatamente esplicite.

I sistemi di monitoraggio disponibili sul mercato sono solitamente strumenti:

- offerti da terze parti, ovvero da fornitori di hardware e software, che hanno soluzioni di monitoraggio compatibili con i loro prodotti
- offerti dai cloud provider, che includono il pacchetto di monitoraggio come parte delle loro soluzioni SaaS
- open source, ovvero soluzioni di monitoring create dalla comunità, che possono essere usate senza pagare alcun costo

Il sistema di alerting (o di allarme), invece, è un meccanismo che genera messaggi specifici ad uno stato del sistema e li recapita ad un determinato destinatario.

L'alerting è un servizio trasversale rispetto al monitoraggio ed è per questo spesso offerto direttamente dai sistemi di monitoraggio. Nella maggior parte dei casi, è perciò importante configurare i criteri di avviso all'interno del sistema di monitoraggio per ricevere notifiche quando si verificano eventi particolari o quando certe metriche violano le regole definite.

Esempi di notifiche da configurare in ottica cloud possono essere:

- violazione di una policy delle attività ammesse sul sistema
- violazione di una policy sui file
- utenza compromessa nel caso ci sia un'alta probabilità che un'utenza sia stata utilizzata in modo non autorizzato
- nuovo utente amministratore
- nuovo luogo di accesso per un amministratore

6.3.9 5.3.9 Automazione

Come anticipato nel capitolo 5.1, le pratiche DevOps[52] rispondono all'interdipendenza tra sviluppo software e IT operations, ed aiutano un'organizzazione a sviluppare e gestire in modo più rapido ed efficiente prodotti e servizi software. L'obiettivo è quello di creare una cultura[53] in cui la consegna del software possa avvenire in maniera veloce, frequente e affidabile, utilizzando l'automazione ove possibile.

L'automazione aumenta la ripetibilità di operazioni critiche e permette di accelerare i processi di delivery, di ridurre al minimo la possibilità di errori o cattive configurazioni aumentando il controllo sui processi. Tutto ciò che viene coinvolto dall'automazione (infrastruttura, ambiente, configurazione, piattaforma, build, test, processo, ecc.) dev'essere definito sotto forma di codice.

I processi che più beneficiano dall'automazione in cloud sono:

- **provisioning dell'infrastruttura:** la grande elasticità dell'infrastruttura messa a disposizione dal cloud, che si traduce ad esempio con la possibilità di scalare ambienti da alcune macchine virtuali a centinaia sulla base del carico, porta con sé un costo operativo e un rischio di errore se legato a processi manuali.

[52] https://cloud.italia.it/glossario/#devops

[53] https://www.digital4.biz/white-paper/forrester-6-tendenze-che-influenzeranno-le-strategie-devops-nei-prossimi-anni_43672157144.htm?__hstc=212895371.9f4238a62197b7846b6fd32be7828223.1552386161803.1553187821467.1553536660708.8&__hssc=212895371.1.1553536660708&__hsfp=916113050

Risulta quindi fondamentale tradurre questi processi sotto forma di codice, così da applicarli una, dieci o mille volte in maniera completamente automatica, creando degli script che permettano di realizzare un ambiente con migliaia di macchine virtuali al mattino e allo stesso tempo spegnerlo alla sera.

È qui che entra in gioco una pratica denominata *infrastructure as code*, che consente appunto di gestire e fare provisioning delle risorse infrastrutturali necessarie tramite file di configurazione processabili automaticamente da strumenti specifici.
I vantaggi principali di questa pratica sono:

- riduzione del rischio di errore umano insito in un processo manuale
- velocità e ripetibilità del processo
- possibilità di fare controllo di versione[54] sul codice che rappresenta l'infrastruttura, così da avere uno storico dei cambiamenti
- possibilità di riutilizzare script e configurazioni in ambienti e progetti diversi
- possibilità di costruire pipeline di automazione che includano anche la parte infrastrutturale
- esistenza di veri e propri registri con template di infrastrutture già pronti
- **distribuzione o deployment:** tramite l'utilizzo di strumenti specifici è possibile automatizzare i processi ripetibili che compilano, impacchettano, distribuiscono e configurano il software dell'applicativo per poi rilasciarlo su ambienti di test o produzione, creando le cosiddette pipelines di build e rilascio.
  I vantaggi di questo approccio e del suo utilizzo in cloud sono:
  - riduzione dei tempi di rilascio del software e di nuove funzionalità o correzioni di bug
  - riduzione del rischio di errore umano insito in un processo manuale
  - possibilità di integrare tutta una serie di test automatici che validino il software sotto vari aspetti, ad esempio funzionale o di sicurezza, per garantire il rilascio solo se tutti i pre requisiti sono rispettati
  - integrazione di strumenti di continuous integration e deployment con i provider cloud
  - possibilità di creare pipelines con servizi completamente gestiti dai provider cloud e con template pre-compilati presenti su marketplace appositi
- **gestione automatica dei disservizi:** la maggior parte dei provider cloud offre diversi servizi per il monitoraggio e logging dell'infrastruttura e degli applicativi, i quali espongono le informazioni sullo stato del sistema attraverso apposite API.
  Questo permette di creare script con policy per la gestione automatica di situazioni critiche. Ad esempio è possibile creare script che reagiscono quando i tempi di risposta di un servizio superano una certa soglia, andando a scalare automaticamente le risorse in modo da distribuire il carico.
  I benefici di questo approccio sono i seguenti:
  - monitoring automatico dei livelli desiderati di servizio con generazione di alert in caso di valori fuori scala
  - possibilità di creare script automatici che implementino azioni correttive o che informino gli amministratori di sistema per poter intervenire tempestivamente
  - velocità di reazione ai problemi e conseguente risoluzione degli stessi
  - possibilità di implementare policy per la gestione della sicurezza

[54] https://it.wikipedia.org/wiki/Controllo_versione

6.3.10 5.3.10 Disaster recovery

Un disastro è una situazione derivante da un evento, naturale o provocato dall'uomo, per via della quale la capacità dell'organizzazione di fornire servizi ai propri utenti è seriamente minacciata o compromessa.

Una strategia di *disaster recovery* deve definire i possibili livelli di disastro e identificare la criticità dei sistemi e delle applicazioni, individuando quali di questi sono più o meno vitali per la salvaguardia delle attività. Un piano di disaster recovery deve stabilire sia misure preventive di sicurezza che misure correttive in caso di emergenza. Inoltre, deve pianificare chiaramente le fasi per ripristinare l'infrastruttura, i sistemi e i dati nel minor tempo possibile e con il minimo sforzo da parte del team.

Il cloud computing porta un approccio completamente diverso al disaster recovery: la virtualizzazione dei server, la disponibilità di data center distribuiti in varie aree geografiche ed i servizi specifici per il disaster recovery permettono un ripristino più rapido dei sistemi IT più importanti senza sostenere le spese di un secondo sito fisico.

I piani di disaster recovery in cloud traggono molti benefici rispetto a quelli tradizionali:

- la virtualizzazione dei server e la containerizzazione degli applicativi permettono di includere tutto ciò che è necessario per il funzionamento in un singolo pacchetto software che può essere copiato, o di cui se ne può fare il backup, in un diverso data center e ripristinato in un tempo dell'ordine dei minuti riducendo significativamente i tempi di ripristino rispetto agli approcci tradizionali
- i sistemi in cloud scalano in modo più semplice, sia in modo verticale che orizzontale
- i costi sono legati all'utilizzo effettivo e non c'è necessità di investimenti iniziali
- l'ampia banda e le alte prestazioni dei dischi per l'I/O solitamente disponibili sulle soluzioni cloud rendono il ripristino più rapido
- possono sfruttare la ridondanza geografica (ovvero la replica dei contenuti su data center distribuiti) offerta dai cloud provider
- i cloud provider offrono i loro servizi da diverse *region* del mondo permettendo la scelta del posto più appropriato per i propri bisogni di disaster recovery

Le diverse strategie di disaster recovery beneficiano significativamente del cloud computing:

1. **backup e restore**: in cui si effettua il backup dei server virtualizzati o degli applicativi containerizzati ed i loro dati su un diverso data center o su una diversa *region* in cloud e, nel caso si verifichi la necessità e in base al disastro verificatosi, si ripristinano sullo stesso data center/region o su uno differente
2. **pilot light**: in cui si mantiene in cloud, in una *region* o data center differente, un ambiente funzionante ma con risorse minime dei componenti più critici (ad es. il database mantenuto allineato attraverso mirroring o replica) e, quando è necessario effettuare il recovery, si procede con il provisioning delle componenti non ancora attive ed a scalare il sistema in modo opportuno per supportare il traffico di produzione
3. **warm standby**: in cui si mantiene in cloud, in una *region* o data center differente, un ambiente pienamente funzionante ma con risorse minime di tutti i componenti e, quando è necessario effettuare il recovery, si procedere a scalare il sistema in modo opportuno per supportare il traffico di produzione
4. **soluzione multi-sito**: in cui si mantiene in cloud, in una region o data center differente, un secondo ambiente pienamente funzionante, dimensionato per supportare il traffico di produzione ed attivo nella ricezione di quest'ultimo

La strategia di *disaster* recovery più opportuna va definita in funzione dei tempi di ripristino desiderati, della quantità di dati che è accettabile poter perdere (RPO[55]) e dei costi da sostenere per adottare le misure necessarie per ottenere tali obiettivi.

[55] https://it.wikipedia.org/wiki/Recovery_Point_Objective

6.3.11 5.3.11 Backup

Al fine di ridurre al minimo le perdite di informazioni in caso di problemi inaspettati durante la migrazione al cloud, deve essere implementata una comprensiva strategia di backup e ripristino, con test periodici del ripristino per assicurarsi che quest'ultimo funzioni correttamente.

La strategia di backup si articola su diversi aspetti, quali:

- l'entità di cui si effettua il backup:
  - backup dell'applicativo, ovvero della sua immagine
  - backup del database, ovvero un backup dello schema del database e dei dati contenuti
- la frequenza temporale:
  - **giornaliero**, ovvero un backup incrementale dell'applicativo, che viene implementato utilizzando strumenti di pianificazione del backup ed eseguito ogni giorno sul cloud
  - **settimanale**, ovvero un backup completo dell'applicativo (inclusivo di tutti i dati del progetto, compresi i repository dei software di gestione), che viene eseguito sul cloud
- la quantità di dati salvati:
  - **backup completo**: la più semplice e completa forma di backup che copia tutti i dati verso un dedicato sistema di memorizzazione. È semplice mantenere il versioning di backup completo, ma il tempo di esecuzione è crescente al crescere della quantità di dati da trattare
  - **backup incrementale**: la copia dei soli dati che sono cambiati dal backup precedente sulla base del timestamp di modifica del file e della data dell'ultimo backup. Un backup incrementale è inferiore in dimensione rispetto ad uno completo e quindi occupa meno spazio nel dispositivo di memorizzazione e richiede minor tempo di esecuzione, per cui può essere pianificato più frequentemente. Al contempo, il recovery del sistema richiede più tempo in quanto tutte gli incrementi dall'ultimo backup completo vanno ripristinati e, se uno dei backup incrementali non è andato a buon fine, il ripristino è incompleto
  - **backup differenziale**: la copia di tutti i file che sono cambiati rispetto all'ultimo backup completo. Questo approccio richiede meno spazio di memorizzazione rispetto al backup incrementale e per il restore sono necessari solo l'ultimo backup completo e quello differenziale, ma il tempo di esecuzione è superiore rispetto al backup incrementale
- il periodo di conservazione dei dati:
  - è importante definire una politica di conservazione dei dati che definisca i tempi minimi di mantenimento dei backup, oltre i quali è opportuno dismettere le informazioni in modo da liberare lo spazio di memorizzazione da loro utilizzato
  - la politica di conservazione deve garantire il ripristino dei dati corretti e della giusta quantità di dati nel sistema in caso di perdita di dati
  - la politica di conservazione deve trattare diversamente l'archiviazione dei dati dal backup dei dati: i dati archiviati non sono più attivamente in uso ma sono necessari per una conservazione di lungo periodo per consultazione o adeguamento alle normative. I dati archiviati sono memorizzati su dispositivi di memorizzazione più economici e devono essere semplici da ricercare
  - per un'appropriata creazione e realizzazione della politica di conservazione il team IT ed il team legale devono collaborare, in quanto il team legale ha maggiore consapevolezza della durata per cui i dati devono essere conservati
  - object storage[56] sono, tra i servizi disponibili in cloud, utilizzati frequentemente per la conservazione sul lungo periodo dei dati in quanto risultano più economici di soluzioni on-premise e garantiscono una migliore protezione dei dati

[56] https://en.wikipedia.org/wiki/Object_storage

  - i dati sono in costante aumento non solo nei dispositivi di memorizzazione primari, ma anche in quelli di backup e di archiviazione. I dispositivi di backup rappresentano in particolare un aggravio economico quando la stessa informazione è salvata più volte. La definizione di una politica di conservazione dei dati è un modo per ridurre la dimensione dei backup ed, eventualmente, automatizzare la rimozione di alcuni insiemi di dati. Tuttavia, insiemi diversi di dati possono avere tempistiche di conservazione diverse, per cui una buona politica, volta ad ottimizzare occupazione e costi, deve considerare anche dove un certo insieme di dati debba essere conservato

Le procedure di backup e restore traggono vantaggio dall'essere effettuate su una piattaforma cloud:

- durabilità dei dati grazie alla ridondanza dei dispositivi di storage
- flessibilità e scalabilità grazie alla possibilità di aumentare le risorse per il backup in pochi minuti
- efficienze di costo grazie alle tariffe a consumo
- sicurezza e compliance grazie al controllo degli accessi, la crittazione e gli strumenti di auditing

Per impostare la desiderata politica di backup è raccomandato l'utilizzo degli strumenti dedicati offerti dal cloud service provider.

6.4 5.4 Gli scenari di migrazione

6.4.1 5.4.1 Virtualizzazione

La virtualizzazione consiste nell'eseguire un'istanza virtuale di un sistema informatico attraverso un livello che astrae l'hardware fisico. Su un unico server fisico (host) è possibile eseguire più sistemi operativi con relative librerie e applicazioni. Questo meccanismo esegue i programmi in una macchina virtuale isolandoli da quelli eseguiti in un'altra macchina virtuale sullo stesso host.

La virtualizzazione permette di applicare la strategia di migrazione di Re-host ("Lift and Shift") indicata nella sezione 4.1.4. Questa strategia è particolarmente adatta a migrare un'applicazione legacy su una piattaforma cloud in quanto, invece di spostare progressivamente e separatamente nel tempo i componenti di un applicativo, sposta l'intero ambiente in un colpo solo, con tutto l'insieme di complesse dipendenze.

Le principali fasi di una migrazione Re-host sono:

1. identificare le macchine virtuali da migrare
2. identificare i prerequisiti come il sistema operativo che può essere supportato nel cloud computing (Linux o Windows)
3. controllare l'elenco dei formati di immagine che sono supportati in cloud (formato raw, VHD, VMDK)
4. per iniziare e condurre la migrazione, è necessario installare gli strumenti cloud-cli (command line interface), tipicamente forniti dal cloud provider, sulla macchina su cui risiedono le immagini di origine
5. preparare le macchine virtuali ed esportare le macchine virtuali dal loro ambiente virtuale
6. prevedere i tempi di downtime, informare tutte le parti interessate e pianificare temporalmente l'attività
7. importare l'immagine della macchina virtuale in cloud e controllare lo stato dell'importazione. Una volta che l'importazione è stata eseguita correttamente, avviare le macchine virtuali in cloud e verificare lo stato dell'istanza

6.4.2 5.4.2 Containerizzazione

La containerizzazione è una strategia di virtualizzazione alternativa a quella tradizionale basata su hypervisor[57]. che si basa su una serie di funzionalità di isolamento messe a disposizione dal sistema operativo ospitante.

Mentre nella virtualizzazione il sistema host emula completamente un nuovo sistema operativo e una serie di periferiche virtuali (disco, network, ..) nella containerizzazione si usano le funzionalità del sistema operativo per creare al suo interno un contesto di esecuzione separato (container) per uno o più processi; questo risulta in un utilizzo più efficiente delle risorse poiché non è necessario emulare un'intera macchina virtuale, ma il livello di isolamento è minore poiché i processi applicativi comunicano con lo stesso kernel[58].

La soluzione open source Docker[59] offre una piattaforma di containerizzazione che può essere una alternativa più leggera agli approcci basati su hypervisor.

La metodologia tipica di delivery di un container si basa sul concetto di immagine, ossia un file contenente tutte le componenti applicative (files, librerie, ..) utili all'esecuzione dei processi associati al container; le piattaforme di containerizzazione mettono a disposizione degli strumenti per la creazione, la pubblicazione e la distribuzione delle immagini.

Questa modalità permette di evolvere i software con maggiore flessibilità rimuovendo ad esempio le dipendenze degli applicativi dall'infrastruttura di middleware: modificata l'immagine, viene istanziato un nuovo container indipendente dagli altri e che comunicherà con essi tramite API e protocolli rete.

I container funzionano su sistemi bare-metal, istanze cloud e macchine virtuali, su Linux, Windows e Mac, permettendo la portabilità di un'immagine da un sistema all'altro solitamente senza variazioni.

I container vengono spesso utilizzati per il deployment di microservizi ed in congiunzione ad uno strato di orchestrazione che si occupa di scalare separatamente il numero dei container associati ai servizi in funzione del carico applicativo della singola componente.

La containerizzazione offre una serie di vantaggi:

- piattaforma multi-cloud: uno dei principali vantaggi dei container è che possono operare su diverse piattaforme di cloud provider, creando una piattaforma multi cloud
- condivisione dello stesso sistema operativo: i container condividono lo stesso kernel del sistema operativo dell'host. Pertanto, i container possono essere più efficienti di una VM che richiede l'esecuzione di un sistema operativo separato
- test e CI / CD: i container mantengono invariato l'ambiente di esecuzione di un'applicazione in tutti i suoi passaggi ad ambienti diversi come tipico per approcci di continuous integration e continuous deploy
- portabilità: i container hanno una portabilità migliore rispetto ad altre tecnologie di hosting. Possono muoversi verso qualsiasi sistema. La configurazione del container è anche portatile in quanto è solo un file da condividere
- versionamento: questa è una delle parti più importanti del ciclo di sviluppo del software. Docker offre il controllo della versione che semplifica il ripristino a un'immagine precedente se si interrompe l'installazione
- costo: i container sono anche economici. Nonostante gli investimenti in memoria, CPU e storage, è possibile supportare molti container sulla stessa infrastruttura
- velocità: i container girano più velocemente delle macchine virtuali, il che è molto importante per le applicazioni distribuite

I container consentono di suddividere le applicazioni in microservizi più piccoli e gestibili. Ogni microservizio è autosufficiente e può essere modificato e aggiornato da solo senza la necessità di toccare gli altri servizi. Ad esempio, se è necessario effettuare un aggiornamento, è necessario modificare e compilare solo i servizi interessati anziché ricompilare l'intera applicazione.

[57] https://it.wikipedia.org/wiki/Hypervisor
[58] https://it.wikipedia.org/wiki/Kernel
[59] https://it.wikipedia.org/wiki/Docker

Kubernetes[60] può essere utilizzato per gestire (tecnicamente detto "orchestrare") questi singoli servizi. Per sfruttare tutti i vantaggi dei container e di Kubernetes, valutare le applicazioni legacy per capire se possono essere suddivise in moduli: non tutte le applicazioni legacy possono essere suddivise in moduli più piccoli.

I seguenti passaggi possono aiutare nella valutazione:

- non aggiungere altro alle applicazioni legacy. Iniziare da zero, riscrivendo anche le applicazioni legacy nell'architettura dei microservizi, è sicuramente una scelta valida, soprattutto se è stata presa la decisione di adottare completamente i container
- se le applicazioni legacy non possono essere scomposte in moduli, la cosa più semplice da fare potrebbe essere semplicemente racchiudere l'applicazione in un unico contenitore
- la semplice suddivisione delle applicazioni in container non le rende automaticamente scalabili. È necessaria una pianificazione adeguata per determinare come devono essere eseguiti questi singoli container. Kubernetes crea container dentro a pod[61] ed offre un DaemonSet[62], che è un modo automatico per creare pod di contenitori man mano che vengono aggiunti più nodi server. L'utilizzo di tali funzionalità per scalare con i microservizi deve essere considerato in anticipo

6.4.3 5.4.3 Ristrutturare l'applicativo

La ristrutturazione di un applicativo per renderlo più adatto a sfruttare le potenzialità delle piattaforme cloud può essere fatta a diversi livelli di profondità:

- riducendo le dipendenze da sistemi esterni
- sostituendo componenti con le versioni cloud native
- riprogettando le strutture interne dell'applicativo e trasformandolo per assumere un'architettura più idonea ad un'efficace erogazione del servizio associato

Qualunque sia lo scenario che motiva la modifica a livello di codice sorgente dell'applicativo è opportuno seguire dei principi moderni di progettazione del software che aiutino quest'ultimo ad essere sempre più adattabile alle continue evoluzioni del bisogno degli utenti e del servizio associato.

In accordo con le La carta dei principi tecnologici del procurement[63] definite dal Codice dell'Amministrazione Digitale e dal Piano Triennale, l'obiettivo è quello di sviluppare servizi che:

- soddisfino le esigenze degli utenti/cittadini
- siano facilmente manutenibili
- siano capaci di evolvere in base alle esigenze dei cittadini e al progresso tecnologico
- siano indipendenti da singole componenti architetturali di terze parti
- diminuiscano le situazioni di dipendenza da un ristretto numero di fornitori (lock-in)

Affinché gli applicativi della Pubblica Amministrazione possano sfruttare i benefici del cloud è necessario che adottino principi di progettazione moderni per:

- ottenere architetture in grado di sfruttare appieno le potenzialità delle piattaforme cloud
- considerare le differenze rispetto alla situazione on-premise

[60] https://kubernetes.io/
[61] https://kubernetes.io/docs/concepts/workloads/pods/pod/#what-is-a-pod
[62] https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/
[63] https://carta-dei-principi-tecnologici-del-procurement.readthedocs.io/it/latest/

5.4.3.1 Basso accoppiamento

Affinché gli applicativi della Pubblica Amministrazione possano sfruttare i benefici del cloud è necessario che adottino architetture moderne in linea con i principi secondo cui le piattaforme cloud funzionano.

Uno delle architetture meno adatte all'uso in cloud è l'architettura monolitica, in cui gli applicativi sono sviluppati e distribuiti come una singola entità e:

- crescono in complessità (n. di dipendenze interne) al crescere della ricchezza funzionale
- richiedono il test dell'intera applicazione per la verifica d'impatto di un cambiamento
- scalano l'intero sistema in modo uniforme anche a fronte di carichi localizzati

Architetture monolitiche non si prestano allo sviluppo di applicativi complessi che devono evolvere rapidamente ed ottimizzare il consumo di risorse e costi rispetto ai carichi da gestire.

Gli applicativi con architettura multi-tier sono nati come evoluzione dei monoliti. Sono composti da diversi strati a livello di stack tecnologico, ad es. nel caso 3-tier: uno strato di dati, uno strato di logica di business e uno di interazione con l'utente con lo scopo di permettere una gestione separata dei livelli riducendo la complessità per ognuno di essi. Nonostante questa divisione, le applicazioni di questo tipo aumentano di complessità con l'aumento della ricchezza funzionale e presentano gli stessi svantaggi dei monoliti in termini di scalabilità.

Per superare i limiti di architetture monolitiche e multi-tier, si è cominciato a scomporre gli applicativi per funzionalità di business, considerandoli una collezione di servizi piuttosto che un unicum. Questo tipo di applicativi hanno un'architettura conosciuta come SOA ("Service Oriented Architecture") che offre vantaggi in termini di:

- scalabilità, in quanto ogni servizio può essere scalato indipendentemente dagli altri
- gestione, in quanto le dimensioni ridotte di ogni servizio rispetto all'applicativo complessivo permettono un alto livello di controllo sul funzionamento o sull'impatto di un cambiamento
- interoperabilità, in quanto ogni servizio espone un contratto (API) con cui altri servizi (interni o esterni) possono utilizzarlo

Architetture ancora più moderne rispetto a SOA sono quelle a microservizi e che utilizzano container. I vantaggi principali di queste architetture sono:

- la definizione di componenti indipendenti e di dimensioni molto-ridotte (micro-servizi) che semplificano il lavoro di più team sullo stesso codice sorgente abilitando l'ownership a livello di micro-servizio, il controllo sugli impatti dei cambiamenti (testabilità), l'ammodernamento attraverso sostituzione di un micro-servizio obsoleto ed un processo di build più efficiente in quanto a livello di singolo micro-servizio
- la definizione di interazione attraverso API RESTful che rendono la realizzazione dei micro-servizi non vincolata all'utilizzo di un unico stack tecnologico e rafforzano la capacità del team di sviluppo a lavorare in parallelo sullo stesso sistema
- l'astrazione rispetto all'ambiente di run-time (container) che riduce le componenti cui l'applicativo dipende direttamente

I servizi si sviluppano e distribuiscono in modo indipendente e sono più facili da manutenere, correggere e aggiornare, garantendo funzionalità più agili per rispondere ai cambiamenti.

Per sfruttare appieno i benefici del cloud, gli applicativi monolitici o multi-tier devono evolvere verso (e i nuovi applicativi devono essere sviluppati con) architetture moderne, da SOA a microservizi.

Le architetture moderne sono caratterizzate da un basso accoppiamento, cioè una tecnica volta a massimizzare l'indipendenza tra i diversi componenti applicativi attraverso l'uso di API.

5.4.3.2 Design for failure

L'approccio "design for failure" richiede di progettare applicazioni in modo che un malfunzionamento dell'applicativo causi solo un degrado proporzionale alla funzionalità che non funziona ma non pregiudichi la fruizione nel complesso dell'applicativo. Secondo questo principio, devono essere rispettate precise linee guida per lo sviluppo e la gestione dell'applicativo:

- sfruttare i meccanismi di fault-tolerance della piattaforma cloud (per approfondimento a riguardo vedi sezione 5.2.1.2 - Disponibilità)
- utilizzare più zone di disponibilità (località fisiche separate offerte dal provider) per proteggere le applicazioni e i dati da eventuali guasti del datacenter
- implementare una strategia di backup e ripristino costante e automatico
- evitare di sincronizzare copie "in-memory" di grandi quantità di dati da uno o più storage centrali all'interno degli applicativi: scalabilità e ridondanza dei sistemi, sono possibili anche grazie alla facilità con cui è possibile creare e distruggere istanze replica dell'applicativo ed in caso di storage "in-memory" la creazione di un'istanza forzerebbe ogni volta una nuova ed onerosa sincronizzazione che impatterebbe a sua volta sulle tempistiche di restore del servizio
- creare e manutenere immagini per macchine virtuali o container che contengano tutte le dipendenze necessarie agli applicativi così da mitigare errori nelle procedure di rilascio dovuti a possibili dipendendenze esterne non più soddisfatte
- configurare un dashboard di monitoraggio che permetta di identificare il punto di malfunzionamento in caso di fallimento o problema di performance

6.5 5.5 La validazione

Come già descritto precedentemente, il processo di migrazione in cloud degli applicativi dell'amministrazione comporterà diversi cambiamenti che potrebbero avere un impatto negativo su vari aspetti come performance, funzionalità o dati degli applicativi migrati.
Risulta quindi fondamentale l'utilizzo di pratiche che permettano di validare i cambiamenti attuati così da ridurre i rischi associati e garantire il corretto funzionamento del sistema.

La validazione è una pratica che viene applicata in maniera continuativa e non solo alla fine di un processo di migrazione, così da accorciare il ciclo di feedback e reagire più velocemente in caso di problematiche.
Nel validare la migrazione di uno o più applicativi vengono considerati diversi aspetti ad essi associati:

- **validazione delle funzionalità applicative**
- **validazione dell'integrità e completezza dei dati** (vedi capitolo 6)
- **validazione del mantenimento o miglioramento delle prestazioni**
- **validazione della sicurezza**

La validazione di tutti questi aspetti passa attraverso l'utilizzo di diverse tipologie di test che vedremo nel corso del capitolo e che variano a seconda dell'aspetto da testare.

6.5.1 5.5.1 Tipologie di testing

Il testing del software può essere eseguito in due modi:

- **manuale:** eseguito da tester che validano manualmente i test case senza utilizzare alcuno strumento di automazione. Il testing manuale è soggetto al rischio dell'errore umano e richiede un maggiore investimento, sia in termini di tempo che di risorse. Tuttavia, questo tipo di testing permette l'osservazione umana, che può essere utile se l'obiettivo è la facilità d'uso o migliorare l'esperienza degli utenti
- **automatizzato:** eseguito utilizzando strumenti di automazione per validare la suite di test case. Il testing automatizzato è più affidabile, in quanto viene eseguito da strumenti e/o script, ed è anche molto più veloce del testing manuale. Può richiedere un investimento per acquisire gli strumenti di testing necessario, ma il costo viene solitamente ammortizzato poiché i test vengono eseguiti più volte nel corso di un lungo periodo di tempo

In generale, ci sono diverse parti e caratteristiche degli applicativi che possono essere testate. Per questo motivo, esistono diverse tipologie di test:

- **unitari:** solitamente testano unità logiche interne al software e create dagli sviluppatori stessi, quindi non visibili ad un utente finale, in maniera isolata da tutte le altre unità logiche che compongono il sistema.
- **di integrazione:** testano l'integrazione del core applicativo con componenti esterne, come ad esempio i database.
- **di interfaccia:** testano il funzionamento dell'applicativo attraverso la sua interfaccia di utilizzo che sia grafica (UI) o programmatica (API)
- **di sistema:** testano il funzionamento end to end dell'applicativo nell'insieme dei suoi componenti
- **funzionali:** testano che l'applicativo rispetti i requisiti funzionali dal punto di vista dell'utilizzatore.

6.5.2 5.5.2 Validazione delle funzionalità

Sostituire un applicativo con una versione SaaS o migrare lo stesso in Cloud sono procedure che potrebbero comportare la perdita o il malfunzionamento di alcune funzionalità precedentemente presenti nella versione on-premise, causando un disservizio più o meno critico ai suoi utilizzatori.

La validazione delle funzionalità punta, attraverso test funzionali, a identificare la presenza o meno di questa tipologia di errori così da poterli correggere.

La lista delle funzionalità desiderate è testata attraverso la verifica di input e di output corrispondenti, senza porre l'attenzione su come i dati vengano processati. I test simulano l'effettivo utilizzo del sistema dal punto di vista di un utilizzatore e possono essere effettuati sia manualmente, attraverso operatori che eseguono i cosiddetti test case, che automaticamente attraverso l'utilizzo di tool specifici che simulano l'operato di un essere umano.

Nel processo di validazione delle funzionalità vanno tipicamente considerati i seguenti passi:

1. identificazione della lista di funzionalità desiderate
2. creazione dei test case tramite la definizione di un input e del relativo output che ci si aspetta, per ogni funzionalità
3. esecuzione del test case
4. verifica di input e output
5. produzione di un report che evidenzi eventuali problematiche

Il risultato di questa validazione potrà così essere utilizzato nelle successive iterazioni, per risolvere eventuali problemi sorti nel corso della migrazione.

6.5.3 5.5.3 Validazione delle prestazioni

Nel caso di migrazione ad applicativi SaaS, garantire alte prestazioni è compito del fornitore del servizio.

In caso di migrazione a servizi PaaS o IaaS invece, spesso si assume che semplicemente spostando un applicativo al cloud, questo potrà automaticamente beneficiare di tutti i suoi vantaggi, come ad esempio la scalabilità, come se il design dell'applicativo fosse già cloud native.

La realtà è che spesso gli applicativi non possono godere appieno del nuovo ambiente, in quanto ad esempio la scalabilità offerta dal cloud non può sopperire alle prestazioni non ottimali dei software stessi.

Questo tipo di problema si manifesta chiaramente quando applicativi che non hanno mai dato problemi on-premise, smettono di funzionare in cloud a causa ad esempio di alti carichi di utilizzo.

Per questo motivo risulta fondamentale validare le prestazioni degli applicativi migrati attraverso l'utilizzo di strumenti di monitoring e attraverso specifici test:

- **di carico:** verificano le prestazioni dell'infrastruttura per un determinato carico di utilizzo ed in un determinato periodo di tempo. Ad esempio: verifica dei livelli di memoria raggiunti con un carico di 1000 utenti simultanei per 5 minuti
- **di stress:** verificano qual è la capacità massima dell'infrastruttura aumentando progressivamente il carico di utilizzo su di essa fino al raggiungimento del limite massimo
- **di picco:** verificano come reagisce l'infrastruttura a carichi molto variabili, abbassando o alzando di molto il carico in un determinato range di tempo
- **di connettività:** verificano tramite test automatici e manuali la latenza[64] dell'applicativo migrato rispetto a quello on-premise

Nel processo di validazione delle prestazioni vanno considerati i seguenti passi:

1. identificazione dei livelli di prestazioni desiderati per metriche come ad esempio:
    a. tempi di risposta
    b. throughput
    c. livelli di utilizzo di risorse:
        i. CPU
        ii. memoria
        iii. storage su disco
        iv. network
2. definizione dei tipi di test da attuare e dei dati da utilizzare (piano di test)
3. configurazione degli strumenti di monitoraggio dei livelli di prestazioni identificati
4. esecuzione dei test tramite strumenti appositi
5. analisi dei risultati e tuning delle risorse

6.5.4 5.5.4 Validazione della sicurezza

Le responsabilità per la sicurezza in cloud ricade fondamentalmente sotto due categorie: quella associata al cloud service provider che dovrà assicurarsi di rendere l'infrastruttura sicura e quella di chi fruisce dei servizi che dovrà assicurarsi che gli applicativi utilizzino tutte le misure necessarie garantire la sicurezza, argomento di cui parleremo in questo sottocapitolo.

Migrare in cloud gli applicativi, apre ad uno spettro di rischi che vanno considerati tramite l'applicazione di alcune pratiche:

[64] https://it.wikipedia.org/wiki/Latenza

- **sicurezza pre-migrazione:** prima di migrare risulta critico eseguire una review di tutti gli account e relativi permessi di accesso ai dati, così da evitare credenziali scadute che potrebbero compromettere la sicurezza una volta in cloud.

  È inoltre molto importante avere una procedura per lo smaltimento delle risorse on-premise, una volta che non saranno più necessarie, avendo cura di eliminare qualunque dato già migrato, se non più necessario
- **identity management:** tra i vari servizi delle più moderne piattaforme cloud, vi è la possibilità di controllare l'accesso ad ogni informazione e risorsa messa a disposizione dal provider. Una gestione così dettagliata permette di essere molto flessibili ma richiede molta attenzione nel tenere traccia di chi può accedere e fare cosa.

  È quindi buona norma consultare la documentazione dei provider cloud che dovrebbero fornire le buone pratiche relative agli strumenti di identity management messi a disposizione
- **test di penetrazione:** questi test, effettuabili tramite strumenti appositi o da fornitori specifici, permettono di effettuare un check di tutte le vulnerabilità più comuni in cloud su più livelli:
  - applicativo: testando le interfacce grafiche (UI) e programmatiche (API)
  - dati: testando il loro accesso tramite applicativo o direttamente da database
  - rete: testando se e quanto la rete protegge l'accesso ai dati
  - normative: verificando quanto l'infrastruttura è conforme alle normative in vigore
- **crittografia del dato:** è importante verificare di avere sempre attivato la crittografia dei dati, sia su database che su storage, così da evitare visibilità di questi ultimi in caso di violazione della sicurezza

CAPITOLO 7

6. Eseguire la migrazione: i dati

La migrazione dei dati è il processo di trasferimento dei dati da un sistema (sorgente) ad un altro (destinazione) utilizzando strumenti e tecniche appropriati. Essendo questo uno degli aspetti più critici di una migrazione al cloud, lo trattiamo approfonditamente in questo capitolo fornendo esempi e buone pratiche specifiche per i diversi scenari che la pubblica amministrazione può trovarsi ad affrontare.

Il processo di migrazione dei dati si articola tipicamente secondo queste fasi:

1. preparazione della migrazione
2. validazione dei dati nel sistema sorgente
3. creazione dello schema dei dati nel sistema destinazione
4. mappatura delle strutture dati del sistema sorgente nel sistema destinazione
5. *conversione e trasferimento dei dati* dal sistema sorgente al sistema destinazione
6. validazione dei dati migrati nel sistema di destinazione
7. dismissione del sistema sorgente

Ognuna di queste fasi può essere eseguita secondo modalità diverse in base al tipo di migrazione.

All'aumentare dei dati memorizzati nel sistema sorgente, il processo di migrazione diventa più complesso ed esposto a rischi come:

- **perdita dei dati**: alcuni dati presenti nel sistema sorgente potrebbero non essere trasferiti correttamente nel sistema di destinazione;
- **inconsistenza dei dati**: anche una migrazione apparentemente corretta può nascondere delle insidie. Un dato potrebbe finire in una colonna differente, o l'ordine di esecuzione delle attività di migrazione potrebbe non rispettare le relazioni tra i dati; Ad esempio dovendo migrare un insieme di utenti e le loro liste dei desideri, le seconde andrebbero migrate dopo i primi;
- **downtime prolungato**: il processo di migrazione può richiedere più tempo di quanto pianificato e durante questo processo il sistema sorgente non è disponibile;

- **corruzione dei dati**: regole di validazione e di codifica differenti tra il sistema sorgente e quello di destinazione possono rendere indisponibili le applicazioni e generare errori anche permanenti, specie se vengono rilevati durante l'utlizzo degli applicativi. Questo può impattare anche sugli utenti finali;
- **interferenze**: se il sistema sorgente o di destinazione sono in uso durante il processo di migrazione, possono verificarsi dei contenziosi sull'accesso e la modifica delle risorse (eg. locking delle tabelle) o un disallineamento dei dati.

Questi rischi possono essere mitigati adottando strumenti e tecniche di test della migrazione dei dati, come raccomandato nella fase di *validazione* (pagina **??**).

7.1 6.1 La preparazione

Prima di iniziare la migrazione vera e propria di un applicativo e dei suoi dati verso una piattaforma in cloud, è necessario considerare una serie di attività preparatorie, in modo da ridurre il rischio di problemi nella transizione che porterebbero ad un conseguente aumento dei tempi per poterla portare a termine.

Le attività preparatorie mirano ad identificare eventuali problematiche da analizzare e risolvere prima di proseguire. La mancata risoluzione di queste potrebbe essere anche motivo per valutare una differente strategia di migrazione.

Quando si vuole migrare un applicativo in cloud è importante seguire le seguenti buone pratiche:

1. fare l'assessment della base dati: la maggior parte dei cloud provider forniscono un tool di assessment per la base dati che supporta la creazione di un report con indicazioni utili alla migrazione, come ad esempio le azioni necessarie per convertire i dati ad un differente schema, informazioni sulle licenze, funzionalità supportate in cloud, configurazione dell'hardware di partenza e una serie di buone pratiche da attuare.
2. assicurarsi che sia possibile fare import ed export di tutti i dati attraverso funzionalità proprie del database di destinazione o strumenti specifici del provider cloud

7.2 6.2 Buone pratiche

Nella migrazione si consiglia di utilizzare le seguenti buone pratiche:

1. prima di procedere con la migrazione eseguire un backup completo del database sorgente
2. trasferire i dati utilizzando la connessione internet se la loro dimensione è ragionevole rispetto alla connessione disponibile; utilizzare una connessione diretta con la piattaforma cloud per grandi quantità di dati;
3. utilizzare una connessione sicura durante il trasferimento dei dati verso il sistema di destinazione in modo da proteggere le informazioni sensibili
4. utilizzare i servizi di migrazione dei dati forniti direttamente dal cloud provider, se disponibili
5. assicurarsi che i dati nel database sorgente non possano essere modificati durante tutto il processo di migrazione e di validazione
6. comparare i tempi di esecuzione di un insieme di query sul sistema sorgente con quelli del sistema di destinazione per capire l'impatto sulle performance
7. suddividere la migrazione di tabelle con grandi quantità di dati in parti più piccole per migliorare le performance nel trasferimento
8. verificare che l'applicazione funzioni correttamente dopo la migrazione

7.3 6.3 Gli scenari di migrazione

Ci sono diversi scenari di migrazione dei dati:

- migrazione verso la medesima versione di sistema di basi di dati
- migrazione verso una versione più recente del sistema di gestione di basi dati rispetto a quello di origine
- migrazione verso un nuovo database, in cui il sistema di destinazione è un software di gestione di basi dati diverso da quello di origine
- cambio di applicativo, in cui i dati devono essere trasferiti ad un diverso applicativo con un diverso schema di dati

La migrazione può essere un processo eseguito una volta sola (one time migration) o in modo incrementale (incremental migration). Il secondo approccio può essere necessario nel caso ci sia la necessità di mantenere pienamente funzionanti sia il sistema sorgente che il sistema destinazione per un periodo di tempo limitato.

7.3.1 6.3.1 Migrazione verso lo stesso sistema di gestione delle basi dati

Nel caso in cui la versione del sistema di gestione della base dati sorgente e destinazione siano il medesimo, è consigliabile utilizzare gli strumenti che il sistema stesso mette a disposizione per la migrazione, come ad esempio lo strumento di dump del database.

Il dump del database sorgente, il suo trasferimento sull'ambiente di destinazione ed il ripristino permettono di completare la migrazione in un'unica volta.

La creazione dello schema del database, la migrazione dei dati e la loro validazione avvengono durante il processo di ripristino.

Utilizzando gli strumenti di backup incrementale del database sorgente, e conseguente ripristino sul database di destinazione, è invece possibile ottenere la migrazione dei dati attraverso un processo incrementale.

7.3.2 6.3.2 Migrazione verso una versione più recente del sistema di gestione delle basi dati

Nel caso in cui la versione del sistema di gestione della base dati di destinazione sia più recente rispetto a quella sorgente, è necessario verificare che vi sia piena compatibilità fra le due versioni in termini di definizione del database prima di procedere alla migrazione vera e propria.

Appurata la compatibilità, è consigliabile utilizzare gli strumenti che il sistema stesso mette a disposizione per la migrazione, come ad esempio lo strumento di dump del database o di backup incrementale come descritto nel sottocapitolo precedente.

7.3.3 6.3.3 Migrazione verso un diverso sistema di gestione delle basi dati

Nel caso in cui si vogliano migrare i dati ad un diverso sistema di gestione delle basi dati, la problematica principale che si deve affrontare è legata alle possibili variazioni nei linguaggi di manipolazione dei dati.

Per gestire la migrazione dei dati è consigliabile utilizzare gli strumenti forniti direttamente dal cloud provider. Se non sono disponibili, si possono utilizzare strumenti di terze parti, anche open source, disponibili sul mercato o servizi che si occupano di fare la migrazione fra un motore di database ed un altro.

La fase di *conversione* dello schema di dati del sistema sorgente nel formato supportato dal sistema di gestione di basi di dati di destinazione *e* la *creazione dello schema* vengono gestite direttamente dallo strumento fornito dal cloud

provider o dallo strumento di terze parti. Per la conversione si raccomanda di utilizzare uno degli strumenti disponibili sul mercato, noti come Schema Conversion Tool[65].

La migrazione incrementale in questo scenario non è raccomandabile in quanto il rischio di corruzione dei dati è molto alto.

7.3.4 6.3.4 Migrazione verso un diverso applicativo

Nel caso in cui si vogliano migrare i dati ad un diverso applicativo si deve tipicamente affrontare un processo di rimappatura delle informazioni dal modello del sistema sorgente al modello del sistema destinazione scrivendo degli script specifici che facciano le trasformazioni necessarie e generino codice che può essere eseguito sul sistema destinazione per caricare i dati modificati.

I principali fornitori di soluzioni SaaS forniscono API per le operazioni di import ed export. In caso non fossero disponibili si suggerisce di verificare la disponibilità sul mercato di strumenti di terze parti.

7.4 6.4 La validazione

L'ultima fase del processo di migrazione deve prevedere i test di accettazione finalizzati a garantire che tutti i dati che si vuole trasferire sono stati effettivamente migrati correttamente.

Esempi di verifiche basilari e non esaustive eseguibili per verificare la consistenza della migrazione sono:

- eseguire le medesime query sul database sorgente e quello di destinazione ed assicurarsi che il risultato sia identico
- verificare che il numero di record nel database sorgente e nel nuovo database sia il medesimo

È raccomandato di adottare anche tecniche avanzate per la validazione della migrazione come quelle descritte di seguito.

7.4.1 6.4.1 Test di completezza

I test di completezza sono finalizzati a verificare che tutti gli oggetti rilevanti a livello business che si vogliono migrare siano stati effettivamente migrati con successo.

I test di completezza si basano sul concetto di Unità di Migrazione (*Unit of Migration, UoM*), ovvero l'identificazione di ogni entità rilevante a livello di business che può tradursi in una o più strutture nella base dati (ad es. tabelle).

Le unità di migrazione sono importanti per capire quali entità non sono state correttamente migrate (ad es. il cittadino i cui dati di anagrafe non sono corretti). In caso di entità articolate su più tabelle si può decidere che, a causa di un fallimento nella migrazione di un record in una qualsiasi tabella, l'intera entità non vada migrata, causando quindi la rimozione dei record legati a quella specifica entità inseriti prima dell'errore (fallout management) o che quell'aspetto venga riportato come errato.

La **riconciliazione** è una tecnica di verifica della completezza che vuole assicurare che il numero di entità nel sistema sorgente sia il medesimo nel sistema di destinazione. Come calcolarlo dipende dallo specifico dominio di appartenenza delle entità che si stanno migrando. Per alcuni domini un semplice conteggio dei record potrebbe essere sufficiente, per altri potrebbe essere necessario eseguire operazioni più complesse, come la cosiddetta validazione orizzontale o la rigenerazione sullo stesso insieme di dati di report e analisi disponibili dal sistema sorgente.

Il conteggio dei record deve considerare gli eventuali duplicati che sono stati rimossi nel processo di migrazione e le entità per cui si sono verificati errori:

[65] https://en.wikipedia.org/wiki/Schema_migration

entità nel database sorgente = entità nel database destinazione + entità rimosse perché duplicati - entità non migrate a causa di errori

Per le entità non migrate a causa di errori deve essere identificato un possibile rimedio.

La **validazione orizzontale** è una tecnica di verifica che si basa sul conteggio di alcuni valori rilevanti delle entità che mira ad assicurare che le trasformazioni, gli arricchimenti di dati, i consolidamenti e le esclusioni non abbiano alterato aspetti fondamentali dell'unità di migrazione. Ad esempio si potrebbe controllare il numero di pagamenti sopra una determinata cifra, o il numero di cittadini per fasce d'età.

Un'altra tecnica che si può utilizzare è **ricreare una serie di report generati nel sistema sorgente nel sistema destinazione e confrontarne il risultato**.

7.4.2 6.4.2 Appearance test

Gli appearance test mirano a ridurre il rischio di inconsistenza dei dati. In questo tipo di test i tester, esperti di dominio, confrontano manualmente i dati presenti nel sistema sorgente e destinazione attraverso l'interfaccia utente dell'applicativo. Un esempio è controllare che le informazioni di residenza riportate sulla scheda utente di un cittadino, siano le stesse tra il sistema on-premise e quello in cloud.

7.4.3 6.4.3 Test di integrazione

Questo tipo di test sono utilizzati quando più applicativi sono interconnessi: quando un applicativo è impattato dalla migrazione, tutti gli applicativi che ne dipendono sono anch'essi impattati. Le tecniche di testing adottate sull'applicativo i cui dati sono stati migrati possono essere utilizzate per validare il funzionamento anche degli applicativi che dipendono da esso.

CAPITOLO 8

7. Dopo la migrazione

Una volta completata la migrazione al cloud di uno o più applicativi, è il momento di riflettere sui risultati raggiunti e sull'impatto generato da questa operazione.

In questo capitolo forniamo un quadro completo degli indicatori di prestazione (key performance indicators o KPI) che possono essere usati (ovvero calcolati e interpretati) per valutare i progressi fatti e il valore ottenuto migrando al cloud.

Inoltre, vengono dettagliati due aspetti pratici da tenere costantemente in considerazione anche al termine di una migrazione: il monitoraggio di soluzioni SaaS aggiunte al Cloud Marketplace e il rispetto degli SLA da parte dei fornitori.

Infine, l'ultimo sottocapitolo di questo documento è dedicato all'importante contributo che le pubbliche amministrazioni possono dare condividendo la loro esperienza, le problematiche emerse e le *lessons learnt* durante il processo di migrazione al cloud.

8.1 7.1 Verifica degli indicatori di performance

I key performance indicators (KPI o indicatori di performance) sono misure che, se utilizzate e interpretate nella maniera corretta, aiutano ad identificare e quantificare il rendimento della migrazione in cloud, dimostrando il suo valore e aiutando a contrastare la resistenza al cambiamento.

Tipicamente le organizzazioni scelgono i KPI per il cloud sulla base dei loro bisogni, identificando criteri operazionali e di business con l'obiettivo di migliorarli. Alcuni esempi possono essere i lunghi tempi di risposta del supporto IT o gli alti costi del servizio di archiviazione.

Una volta identificati i criteri su cui migliorare, è importante definire gli obiettivi in termini di misure di miglioramento. Nel caso degli esempi fatti sopra:

- Di quanto si vuole ridurre i tempi di risposta del supporto IT?
- Quanto si vuole risparmiare sui costi del servizio di archiviazione?

Chiariti i criteri su cui migliorare e gli obiettivi da raggiungere, è possibile definire KPI misurabili di cui tenere traccia. È importante eseguire una prima misurazione dei KPI, così da stabilire una linea base da cui partire, per poi monitorarli nel tempo in maniera continuativa.

Vi sono diversi KPI validi per monitorare l'efficacia della migrazione in cloud degli applicativi e di seguito ne elencheremo alcuni.

8.1.1 7.1.1 Indicatori di risultato

Qual è il costo di avere tutto on-premise? E quali sono le aspettative finanziarie legate alla migrazione in cloud? I seguenti KPI puntano a dare una risposta a queste domande.

7.1.1.1 Costi on-premise

Una migrazione al cloud rappresenta anche il passaggio da un modello di costi CAPEX[66] a OPEX[67]. Quindi, al fine di confrontare i costi del cloud con le spese on-premise, è necessario confrontarli convertendo il TCO del proprio data center (vedi capitolo 3.3) in un costo operativo.

- **costo dell'hardware locale**: spesa per server e reti
- **durata dell'hardware**: quanto spesso è necessario aggiornare o sostituire l'infrastruttura IT. Ciò fornisce la base per il calcolo del costo mensile o annuale per l'acquisto di hardware
- **costo totale del capitale investito**: il costo complessivo dell'aver pagato in anticipo l'attrezzatura, contabilizzando gli interessi che si sarebbero guadagnati se si fosse trattenuto quel capitale o gli interessi e le commissioni pagate per ottenere un prestito
- **costi strutturali**: costi associati all'elettricità per far funzionare l'attrezzatura e fornire spazio, come affitto, attrezzature antincendio e personale di sicurezza
- **personale IT**: il cloud può ridurre il sovraccarico operativo sul reparto IT, liberando il personale permettendo loro di concentrarsi maggiormente sulla creazione e sulla manutenzione di applicativi e servizi. La spesa attuale è più legata alla gestione o allo sviluppo software? Assicurandosi di avere una ripartizione chiara è possibile valutare l'impatto del cloud sul carico di lavoro
- **costi IT in proporzione a tutte le spese dell'ente**: è utile per confrontare le spese IT rispetto alla spesa nel suo complesso

7.1.1.2 Costi di migrazione

Come nel caso del calcolo dei costi on-premise, è necessario stimare ciascuno dei seguenti KPI al fine di tradurli in costi operativi cloud. Ma è anche importante tenere a mente costi di sviluppo, gestione, formazione e del personale se si continuano ad ospitare le applicazioni on-premise.

- **costo delle strategie di migrazione delle applicazioni**: il costo associato al tipo di strategia di migrazione adottata per far funzionare le applicazioni nel cloud (vedi capitolo 4.1)
- **tempi di gestione**: quanto tempo di gestione viene speso per pianificare e supervisionare il processo di migrazione
- **costo della formazione**: il costo della formazione del personale per i nuovi sistemi e tecnologie

7.1.1.3 Costi operazionali del cloud

La riduzione delle spese operative è uno degli obiettivi principali dell'adozione del cloud. È perciò importante monitorare, rivedere e ottimizzare regolarmente i costi del cloud.

[66] https://it.wikipedia.org/wiki/Capex
[67] https://it.wikipedia.org/wiki/Opex

- **fatture mensili**: i costi sono in linea con le previsioni? Vi sono opportunità di riduzione dei costi che potrebbero ridurre ulteriormente l'importo delle fatture? I costi cloud aumentano o diminuiscono?
- **costi del personale**: l'adozione del cloud ha influito sulle spese generali del personale? Ad esempio, l'amministrazione sta spendendo più tempo ad analizzare i costi IT rispetto a prima?
- **costi del personale IT**: la spesa è aumentata o diminuita? In particolare, si sta spendendo di più per le operazioni o meno?
- **costo di partner e strumenti di terze parti**: quali servizi e strumenti di terze parti vengono utilizzati per aiutare a gestire il cloud? Quanto costano? Stanno aiutando a risparmiare denaro, aggiungere valore o generare entrate?
- **costi legati all'energia elettrica**: quanti soldi si stanno risparmiando sull'elettricità per alimentare server, sistemi di archiviazione, raffreddamento e apparecchiature ausiliarie?
- **costi IT in proporzione a tutte le spese aziendali**: sono aumentati o diminuiti rispetto al benchmark on-premise?
- **scostamento del budget**: la spesa per il cloud è allineata al budget? Le cifre di bilancio sono realistiche? Vi è necessità di rivederle?

7.1.1.4 Redditività e flusso di cassa

L'IT è solo uno dei numerosi fattori che possono influire sulla redditività. Attraverso diversi KPI sarà possibile valutare il reale impatto finanziario della migrazione in cloud sulle prestazioni dell'amministrazione.

- **crescita degli utenti**: la base di cittadini che utilizza i servizi sale o scende? È possibile dimostrare una correlazione tra la crescita dell'adozione digitale rispetto alla migrazione in cloud?
- **entrate**: le applicazioni che prevedono la riscossione di entrate per l'amministrazione stanno dando riscontri migliori ora che sono migrate in cloud?
- **risparmi**: le applicazioni in cloud stanno facendo risparmiare soldi all'amministrazione?

8.1.2 7.1.2 Indicatori di impatto

Oltre agli indicatori di risultato, vanno anche considerati i benefici più ampi del cloud, come l'agilità e le migliori prestazioni delle applicazioni, che forniscono valore a lungo termine all'amministrazione.

7.1.2.1 Governance

Il cloud è un ambiente complesso e dinamico. È incredibilmente facile creare nuove risorse infrastrutturali per i servizi dell'amministrazione ma è altrettanto facile perdere di vista la loro esistenza senza una buona governance del cloud atta mantenere sotto controllo i costi e la sicurezza.

- **visibilità sul cloud**: vi è piena visibilità sull'inventario cloud? È noto dove si stanno spendendo i soldi? Esiste un quadro completo delle misure di sicurezza?
- **controllo dei costi**: si sta facendo un uso efficiente dell'infrastruttura cloud? È possibile identificare le risorse inutilizzate e sottoutilizzate?
- **precisione dei rapporti**: i rapporti sui costi sono più accurati? Quanto spesso è necessario correggere gli errori nei rapporti?
- **cicli di audit interno**: ci vuole più o meno tempo per controllare i costi IT?
- **conformità**: il cloud soddisfa gli obiettivi di conformità? È possibile ospitare carichi di lavoro che trattano dati sensibili?

7.1.2.2 Agilità e prestazioni

Un miglioramento del servizio IT e della sua gestione non solo dimostrano che la migrazione in cloud sta producendo risultati tangibili ma è anche un chiaro segno che lo staff è motivato, condivide la visione e ha compreso il valore del cloud.

- **obiettivi di prestazione**: quali sono gli obiettivi di prestazioni IT? Le prestazioni delle applicazioni sono in linea con le aspettative? Ed i il numero di incidenti, guasti e richieste di modifica? Quanto spesso è necessario implementare correzioni di codice e miglioramenti delle applicazioni?
- **disponibilità del servizio**: l'ambiente cloud è più robusto? Quante interruzioni di servizio si sperimentano? E quanto durano?
- **impatto sul mercato**: quali nuovi servizi sono stati sviluppati? Si assiste ad un time to market più veloce?
- **open source**: si sfruttano le tecnologie open source? Riducono il carico di lavoro dei team di sviluppo e operations?

7.1.2.3 Automazione

I processi automatizzati riducono il carico di lavoro manuale sulla gestione dell'infrastruttura, aiutano a rimanere in controllo di infrastrutture complesse e dinamiche e forniscono risposte immediate ad eventi come modifiche alla configurazione o improvvisi aumenti del carico di lavoro.

- **ottimizzazione delle risorse**: si stanno acquistando risorse infrastrutturali dimensionate correttamente rispetto ai bisogni? È possibile ridimensionarle a seconda del carico? È necessario mantenere le risorse attive durante le ore non di punta? È possibile sfruttare l'automazione per eseguire automaticamente queste funzioni?

7.1.2.4 Soddisfazione degli utenti dei servizi

Modernizzando il reparto IT, rendendo efficienti i costi e rinnovando le applicazioni legacy, è possibile ribaltare i benefici del cloud sugli utilizzatori degli applicativi attraverso una migliore esperienza utente e funzionalità prima non attuabili.

- **latenza delle applicazioni**: la velocità con cui gli applicativi in cloud rispondono alle interazioni degli utenti, rispetto alla controparte on-premise.
- **ticket al servizio di supporto**: vi sono meno richieste grazie alla stabilità e scalabilità del cloud?
- **tempo di risoluzione dei ticket**: in cloud vi è la possibilità di operare sulla propria infrastruttura in maniera più avanzata. È importante misurare l'impatto di questo cambiamento sull'operatività del supporto tecnico
- **sondaggi sulla soddisfazione degli utenti**: le risposte indicano eventuali problemi nel servizio?

8.2 7.2 Monitoraggio di nuove soluzioni SaaS aggiunte al Cloud Marketplace

Come illustrato più volte in questo documento, il principio "Cloud First" nel contesto del programma di abilitazione al Cloud delle PA[68], prevede che le pubbliche amministrazioni che si apprestano a migrare i propri applicativi in cloud valutino in prima istanza la presenza di servizi SaaS nel catalogo dei servizi cloud qualificati da ACN (Cloud Marketplace).

Lo sviluppo del mercato dei prodotti software verso la modalità SaaS, infatti, offre un costante aumento di soluzioni in cloud che possono rimpiazzare software precedentemente disponibile solo on-premise con la corrispondente versione

[68] https://cloud.italia.it/programma-abilitazione-cloud/

cloud-native realizzata dal medesimo produttore o con soluzioni equivalenti o migliorative proposte da nuovi soggetti. Adottando, ove possibile, la strategia di Sostituzione o Re-purchase, presentata in dettaglio nel capitolo 4.1.3, le pubbliche amministrazioni possono godere dei benefici del cloud da subito con ridotti costi iniziali.

All'interno del Cloud Marketplace[69] è possibile ricercare i servizi SaaS qualificati e visualizzarne la scheda tecnica che mette in evidenza le caratteristiche tecniche, il modello di costo e i livelli di servizio dichiarati dal fornitore in sede di qualificazione.

Poiché il mercato è in continua evoluzione ed il Cloud Marketplace è continuamente in aggiornamento, si raccomanda di monitorare attivamente e con cadenza regolare questo catalogo per assicurarsi che le nuove soluzioni aggiunte vengano considerate per la migrazione dei servizi ancora non prioritizzati (vedi capitolo 3.2 per il processo di prioritizzazione).

8.3 7.3 Segnalazione di violazioni di SLA da parte dei fornitori qualificati

Gli SLA (service level agreement, ovvero accordi sul livello di un servizio) sono un elemento importante di qualsiasi contratto con i fornitori di servizi in quanto descrivono il livello di qualità e supporto minimo atteso, definiscono le metriche con cui misurarlo e le azioni da intraprendere nel caso i livelli non vengano rispettati. Come illustrato nel capitolo 4.2, in particolare nel caso di fornitura di servizi cloud, ci sono diversi aspetti da considerare e specifici requisiti assicurati dai fornitori qualificati da AGID.

Una volta selezionato il fornitore qualificato (cloud service provider o CSP) con i rispettivi SLA ed eseguita la migrazione, è importante che l'amministrazione si assicuri che il livelli di servizio siano soddisfatti e che, in caso di violazioni, queste siano opportunamente segnalate ad AGID. Per maggiori dettagli sulle condizioni di revoca della qualificazione ai CSP si faccia riferimento alla Circolare n.2 del 9 Aprile 2018[70] per i servizi IaaS e PaaS e alla Circolare n.3 del 9 Aprile 2018[71] per i servizi SaaS. È quindi importante che le amministrazioni pubbliche facciano il monitoraggio di quanto acquisito per vedere che corrisponda a quanto dichiarato e acquisito.

8.4 7.4 Condividere l'esperienza di migrazione al cloud

Come illustrato nel capitolo 2, si prevede che l'amministrazione sia accompagnata durante il processo di migrazione (sin dalle fasi iniziali di prioritizzazione e assessment degli applicativi) dal centro di competenza. Il ruolo del centro di competenza, infatti, è quello di soggetto aggregatore di conoscenza impegnato sia nel ruolo di *advisor* per la specifica amministrazione nel suo territorio, che nella raccolta di *know how* ed esperienze di migrazione da condividere con l'unità di controllo e da rendere poi disponibili per le altre amministrazioni.

Per questo motivo, l'amministrazione dovrà assicurarsi di avere scambi continui (ovvero *feedback loops*) con il centro di competenza durante tutto il processo di migrazione. In particolare, dopo la migrazione, è fondamentale che l'amministrazione effettui una retrospettiva (vedi capitolo 5.1.5 per un approfondimento sulla pratica) includendo il centro di competenza e cercando di evidenziare le problematiche emerse, le soluzioni adottate e le *lessons learnt* durante le varie fasi del processo.

Inoltre, l'esistenza di uno spazio di discussione dove i team delle amministrazioni possano dialogare e condividere le proprie esperienze è un prerequisito per la creazione (e la continua evoluzione) di una solida base di conoscenza sulla migrazione al cloud nell'ambito delle pubbliche amministrazioni.

[69] https://www.acn.gov.it/portale/catalogo-delle-infrastrutture-digitali-e-dei-servizi-cloud

[70] https://trasparenza.agid.gov.it/moduli/downloadFile.php?file=oggetto_allegati/181151234430O__OCircolare+2-2018_Criteri+per+la+qualificazione+dei+Cloud+Service+Provider+per+la+PA.pdf

[71] https://trasparenza.agid.gov.it/moduli/downloadFile.php?file=oggetto_allegati/181151237210O__OCircolare+3-2018_Criteri+per+la+qualificazione+di+servizi+SaaS+per+il+Cloud+della+PA+%28002%29.pdf

Per questo motivo, contatta Developers Italia[72], la comunità che si occupa anche di DevOps per il Cloud delle PA, e Forum Italia[73], dove è possibile condividere i propri commenti e feedback sul Cloud Enablement Kit.

[72] https://developers.italia.it/it/contatti
[73] https://forum.italia.it/c/piano-triennale/data-center-e-cloud