

# **Linee Guida su acquisizione e riuso di software per le pubbliche amministrazioni**

*Release Finale*

**Agenzia per l'Italia Digitale**

**27 lug 2018**



---

## Indice dei contenuti

---

<b>1</b>	<b>Premessa</b>	<b>1</b>
1.1	Finalità e struttura del documento	1
1.2	Software oggetto di queste linee guida	2
1.3	Riuso del software	3
1.4	Soggetti destinatari	3
1.5	Titolarietà	3
1.6	Conformità del <i>software</i> alla normativa	4
1.7	Glossario	5
<b>2</b>	<b>Linee Guida sull'acquisizione di software</b>	<b>7</b>
2.1	Introduzione e contesto normativo	7
2.2	Oggetto della valutazione	8
2.3	Valutazione comparativa	8
2.3.1	Descrizione delle soluzioni	8
2.3.2	Descrizione dei criteri per la valutazione	9
2.3.3	Descrizione delle macro-fasi	10
2.4	Macro fase 1: Individuazione delle esigenze	11
2.4.1	Fase 1.1: Analisi del fabbisogno	12
2.4.2	Fase 1.2: Individuazione dei vincoli	12
2.4.3	Fase 1.3: Redazione del documento descrittivo delle esigenze	12
2.5	Macro fase 2: Analisi delle soluzioni a riuso delle PA e delle soluzioni Open Source	13
2.5.1	Fase 2.1: Ricerca soluzioni riusabili per la PA	15
2.5.2	Fase 2.2: Valutazione soluzioni riusabili per la PA	15
2.5.3	Fase 2.3: Approvvigionamento della soluzione riusabile per la PA	16
2.5.4	Fase 2.4: Ricerca soluzioni Open Source	16
2.5.5	Fase 2.5: Valutazione soluzioni Open Source	17
2.5.6	Fase 2.6: Approvvigionamento della soluzione Open Source	17
2.5.7	Fase 2.7: Accertamento impossibilità	17
2.6	Macro fase 3: Analisi delle altre soluzioni	17
2.6.1	Fase 3.1: Ricerca soluzioni proprietarie	18
2.6.2	Fase 3.2: Studio realizzazione ex novo	19
2.6.3	Fase 3.3: Comparazione soluzioni proprietarie e realizzazione ex novo	19
2.6.4	Fase 3.4: Approvvigionamento soluzione proprietaria o realizzazione ex novo	20
2.7	Total Cost of Ownership (TCO)	20
2.8	Scelta della modalità di erogazione del software	21
<b>3</b>	<b>Linee Guida sul riuso del <i>software</i> (art. 69)</b>	<b>23</b>
3.1	Introduzione e contesto normativo	23
3.2	Modello di riuso	23
3.3	Developers Italia e la ricerca di <i>software</i> in riuso	24
3.4	Processo di messa a riuso del <i>software</i> sotto licenza aperta	24

3.4.1	Scelta di uno strumento di code hosting	25
3.4.2	Registrazione del <i>software</i> aperto su Developers Italia	25
3.4.3	Responsabilità connesse al rilascio	26
3.5	Licenze aperte e scelta di una licenza	26
3.5.1	Contesto	26
3.5.2	Licenze per il <i>software</i> aperto	26
3.5.3	Scelta di una licenza	27
3.6	Rilascio di <i>software</i> esistente sotto licenza aperta	27
3.7	Sviluppo di <i>software</i> ex novo	28
3.7.1	Rilascio di nuovo <i>software</i> sotto licenza aperta	28
3.7.2	Acquisizione della titolarità di <i>software</i> sviluppato ex novo	29
3.8	Manutenzione di un <i>software</i> da parte dell'amministrazione titolare	29
3.8.1	Titolarietà del codice sviluppato in fase di manutenzione	29
3.8.2	Rilascio sotto licenza aperta delle modifiche	29
3.8.3	Supporto alle amministrazioni che riusano	30
3.8.4	Software non ancora rilasciato sotto licenza aperta	30
3.9	Riuso di un <i>software</i> o utilizzo di un <i>software</i> Open Source	30
3.9.1	Utilizzo di <i>software</i> a riuso o Open Source	30
3.9.2	Modifiche ad un <i>software</i> a riuso o Open Source	30
<b>A</b>	<b>Allegato A: Guida alla pubblicazione di <i>software</i> come open source</b>	<b>33</b>
A.1	Premessa	33
A.2	Individuazione dello strumento di code hosting	34
A.3	Scelta della licenza	35
A.4	Attribuzione della licenza ed individuazione della titolarità	35
A.5	Individuazione dei materiali da rilasciare	35
A.6	Rilascio del codice e organizzazione del repository	35
A.7	File README	36
A.8	Documentazione	37
A.9	Tempi di rilascio	37
A.10	Sicurezza	38
A.11	Registrazione del repository su Developers Italia	38
<b>B</b>	<b>Allegato B: Guida alla manutenzione di <i>software</i> open source</b>	<b>39</b>
B.1	Obbligo di rilascio	39
B.2	Obblighi relativi alla manutenzione di <i>software</i> per il quale l'Amministrazione disponga già di un repository	40
B.2.1	Aggiornamento delle dipendenze	40
B.2.2	Descrizione del ruolo di maintainer	40
B.2.3	Interazione sul repository/issue tracker	40
<b>C</b>	<b>Allegato C: Guida alle licenze Open Source</b>	<b>43</b>
C.1	Versionamento delle licenze	43
C.2	Public Domain	43
C.3	Licenze non-copyleft	43
C.4	Licenze copyleft	44
C.5	Licenze copyleft - per le librerie	44
C.6	Licenze non comprese nella classificazione appena introdotta	44
C.7	Le licenze Creative Commons	45
C.8	Licenza applicabile alla documentazione e allegati del software	45
C.9	Compatibilità tra le licenze	45
<b>D</b>	<b>Allegato D: Guida alla presa in riuso di <i>software</i> open source</b>	<b>47</b>
D.1	Modifica di <i>software</i> open source adottato in riuso	47
D.2	Modifica del codice sorgente	48
D.3	Interazione con il maintainer del progetto originale	48
D.4	Pubblicazione di codice open source non originato nel originato nel contesto della PA	49
<b>E</b>	<b>Allegato D: Guida alla presa in riuso di <i>software</i> open source</b>	<b>51</b>

E.1	Modifica di <i>software</i> open source adottato in riuso . . . . .	51
E.2	Modifica del codice sorgente . . . . .	52
E.3	Interazione con il maintainer del progetto originale . . . . .	52
E.4	Pubblicazione di codice open source non originato nel originato nel contesto della PA . . . . .	53
<b>F</b>	<b>Allegato E: Tabella Sinottica degli elementi necessari al percorso decisionale</b>	<b>55</b>



La redazione del documento è stata curata dal gruppo di lavoro istituito con determina 237/2017, a composizione mista Agenzia per l'Italia Digitale e Team per la Trasformazione Digitale:

- Giovanni Bajo, Team per la Trasformazione Digitale
- Viviana De Paola, AgID - Area Trasformazione Digitale
- Daniela Intravaia, AgID - Coordinamento Attività Internazionali
- Guido Pera, AgID - Area trasformazione Digitale
- Umberto Rosini, AgID - Area architetture, standard e infrastrutture
- Guido Scorza, Team per la Trasformazione Digitale.

## 1.1 Finalità e struttura del documento

Le presenti linee guida sono adottate in attuazione dagli articoli 68 e 69 del *Codice dell'Amministrazione Digitale* (di seguito CAD):

- come disposto dall'articolo 68, comma 1 ter, individuano nel capitolo 2 *Linee Guida sull'acquisizione di software* le modalità e i criteri con i quali un'amministrazione deve effettuare la valutazione comparativa descritta nel citato articolo per decidere la modalità di acquisizione di un software.
- come statuito dall'articolo 69, comma 2bis, individuano nel capitolo 3 *Linee Guida sul riuso del software (art. 69)* la piattaforma per la pubblicazione di codice sorgente sotto licenza aperta e documentazione del *software* messo a riuso dalle amministrazioni, indicando anche le modalità tecniche di utilizzo.

Inoltre sostituiscono la precedente circolare 63/2013, intitolata «*Linee guida per la valutazione comparativa prevista dall'art. 68 del D.Lgs. 7 marzo 2005, n. 82 Codice dell'Amministrazione digitale*» e relativi allegati.

Questo documento e la metodologia in esso descritta sono da intendersi come ausilio a un percorso decisionale che rimane sotto la piena responsabilità delle amministrazioni, sia nel momento in cui condividano le soluzioni sia quando le adottino in riuso nel rispetto della normativa vigente, in particolare in materia di pubblica amministrazione digitale, contratti pubblici e protezione dei dati personali. Con riferimento a quest'ultimo settore del diritto, il Regolamento UE 2016/679 ha definito/precisato principi e criteri particolarmente rilevanti rispetto all'oggetto delle presenti linee guida. Tra detti principi e criteri, si evidenzia l'esigenza di considerare la protezione dei dati fin dalla progettazione e per impostazione predefinita (art. 25 Regolamento citato). Non andrà trascurato, altresì, il complesso delle regole tecniche di AgID che possono avere incidenza sulla materia, quali le *Misure Minime di Sicurezza* (circolare 2/2017) e le *Linee guida per lo sviluppo del \*software\* sicuro*.

Il presente documento si pone come punto di avvio di un processo culturale che veda le PP.AA. protagoniste di un sempre maggiore ricorso al *software* aperto, come si ricava dalla lettura del primo comma dell'art. 69, che impone alle pubbliche amministrazioni «che siano titolari di soluzioni e programmi informatici realizzati su specifiche indicazioni del committente pubblico» di «rendere disponibile il relativo codice sorgente, completo della documentazione e rilasciato in repertorio pubblico **sotto licenza aperta...**»

Da tale norma si è preso, pertanto, l'avvio per la predisposizione delle presenti *Linee guida*, avendo individuato nella norma appena richiamata il forte impulso del legislatore all'utilizzo sempre maggiore del *software* di tipo aperto da parte delle pubbliche amministrazioni. Ne sono prova la contestuale eliminazione della previsione del c.d. «catalogo del riuso», senza che ciò impedisca, eventualmente, alle PPAA di sottoscrivere accordi (ad es., in base all'art. 15 della Legge 241/90), per il riutilizzo di soluzioni che non siano conformi al dettato dell'art. 69 co. 1 e che non possano rientrare nelle fattispecie qui trattate, che, si ribadisce, devono essere quelle sottoposte a licenza aperta.

In ogni caso, il legislatore, adottando tale ottica di forte propensione all'open source per le PPAA, ha ragionevolmente previsto una **esclusione generale per le sole «motivate ragioni di ordine e sicurezza pubblica, difesa nazionale e consultazioni elettorali»** - all'art. 69 co. 1 ultimo inciso - , al fine di salvaguardare quei settori più delicati del governo digitale del Paese, che dalla condivisione e gestione comunitaria del SW di tipo aperto potrebbero sentirsi esposti a qualche rischio.

L'approccio sopra descritto, di favore per l'open source, si desume con chiara evidenza anche dal dettato del comma 2 dell'art. 69, che impone alle amministrazioni pubbliche «nei capitoli o nelle specifiche di progetto» di essere «sempre titolar[i] di tutti i diritti sui programmi e i servizi delle tecnologie dell'informazione e della comunicazione, appositamente sviluppati per ess[e]».

Anche in questo caso è prevista una salvaguardia, per le sole ipotesi in cui «ciò risulti eccessivamente oneroso per comprovate ragioni di carattere tecnico-economico».

Di conseguenza, l'art. 68 viene letto e vi viene data attuazione nel presente documento, in piena coerenza con la predetta lettura dell'art. 69.

L'amministrazione pubblica è il soggetto che meglio conosce le proprie esigenze ed è in grado di declinare la metodologia qui proposta, in coerenza sia con il proprio contesto, sia con le caratteristiche dell'acquisizione da effettuare.

In tal senso, le *Linee guida* si pongono non come mero strumento regolatorio, bensì come suggerimento per nuovi processi di accompagnamento, sensibilizzazione ed informazione.

## 1.2 Software oggetto di queste linee guida

Al fine di fugare eventuali dubbi interpretativi, nel contesto degli articoli 68 e 69 del CAD le espressioni «programmi», «soluzioni», «programmi informatici» e «soluzioni ICT» sono da intendersi come equivalenti. L'oggetto dell'obbligo sancito dalla disposizione in commento è il «software». Un elenco non esaustivo quindi di software oggetto di queste linee guida è il seguente:

- Applicazioni web (frontend e backend)
- Applicazioni desktop
- Applicazioni mobile
- Componenti e applicazioni semilavorate
- Framework
- Librerie
- Plugin
- Sistemi operativi
- Siti web (frontend e backend)

È auspicabile che le presenti linee guida favoriscano la razionalizzazione delle soluzioni utilizzate nei settori/servizi comuni alle pubbliche amministrazioni, quali, ad esempio, gestione del personale, gestione



e conservazione documentale, gestione dei processi decisionali, comunicazione istituzionale e trasparenza amministrativa.

Inoltre, è importante notare che il termine «software», come usato in questo documento, non designa solo il mero codice sorgente e/o l'eseguibile, ma anche tutti gli artefatti prodotti durante il processo di sviluppo e utilizzo del software, cioè documentazione, asset grafici, manuali, ecc., così come esplicitato nel comma 1 dell'articolo 69.

## 1.3 Riuso del software

Si intende come «riuso» di un *software* il complesso di attività svolte per poterlo utilizzare in un contesto diverso da quello per il quale è stato originariamente realizzato, al fine di soddisfare esigenze simili a quelle che portarono al suo primo sviluppo. Il prodotto originario viene «trasportato» nel nuovo contesto arricchendolo, se necessario, di ulteriori funzionalità e caratteristiche tecniche che possono rappresentare un «valore aggiunto» per i suoi utilizzatori.

Dal combinato disposto degli articoli 68 e 69 del CAD, il *software* in riuso è esclusivamente quello rilasciato sotto licenza aperta da una pubblica amministrazione. Questo è dunque un sottoinsieme di tutto il *software open source* disponibile per l'acquisizione. Le presenti linee guida distinguono, ove necessario, le modalità di acquisizione di *software* di pubbliche amministrazioni assoggettato a licenza aperta rispetto a *software open source* di terzi.

Un aspetto fondamentale del riuso nel contesto della Pubblica Amministrazione è che l'Amministrazione che «riusa» riceve il software gratuitamente dall'Amministrazione cedente, e lo acquisisce sostenendo solo le spese di suo adattamento, ma non quelle di progettazione e realizzazione.

## 1.4 Soggetti destinatari

I soggetti destinatari delle presenti linee guida sono le pubbliche amministrazioni di cui all'articolo 1, comma 2, del decreto legislativo 30 marzo 2001, n. 165, nel rispetto del riparto di competenza di cui all'articolo 117 della Costituzione, ivi comprese le autorità di sistema portuale, nonché le autorità amministrative indipendenti di garanzia, vigilanza e regolazione, ossia «*gli istituti e scuole di ogni ordine e grado e le istituzioni educative, le aziende ed amministrazioni dello Stato ad ordinamento autonomo, le Regioni, le Province, i Comuni, le Comunità montane, e loro consorzi e associazioni, le istituzioni universitarie, gli Istituti autonomi case popolari, le Camere di commercio, industria, artigianato e agricoltura e loro associazioni, tutti gli enti pubblici non economici nazionali, regionali e locali, le amministrazioni, le aziende e gli enti del Servizio sanitario nazionale, l'Agenzia per la rappresentanza negoziale delle pubbliche amministrazioni (ARAN) e le Agenzie di cui al decreto legislativo 30 luglio 1999, n. 300 e al CONI (per quest'ultima amministrazione fino alla revisione organica della disciplina di settore)*».

Non si applicano le disposizioni sul riuso delle soluzioni ove sussistano «*motivate ragioni di ordine e sicurezza pubblica, difesa e sicurezza nazionale e consultazioni elettorali*».

Con riferimento all'ambito di applicazione delle presenti Linee guida, si auspica l'utilizzo da parte delle PP.AA. degli strumenti di cooperazione e collaborazione offerti dalla normativa vigente, quali gli accordi di collaborazione previsti dall'art. 15 L. 241/1990, al fine di realizzare esperienze di co-progettazione, ampliando la condivisione della conoscenza, di processi decisionali e di percorsi comuni, quali ad esempio centri di competenza e di supporto al ciclo di vita del software.

## 1.5 Titolarità

Ai sensi dell'articolo 69 del CAD in materia di riuso, un'amministrazione deve considerarsi titolare di un *software* realizzato su proprie specifiche indicazioni ogni qualvolta che:

- abbia commissionato tale soluzione - attraverso un contratto di appalto o altra fattispecie negoziale avente analoga natura, anche a prescindere dal titolo dato al contratto - e il **contratto preveda l'acquisizione in**

**capo ad essa di tutti i diritti di proprietà intellettuale e industriale** sul *software* sviluppato su committenza della Pubblica amministrazione, o comunque non attribuisca tale titolarità in capo all'appaltatore o a terzi.

- il *software* sia stato realizzato da risorse interne all'amministrazione stessa.

Ogni amministrazione deve, in sede di negoziazione di un contratto volto a commissionare lo sviluppo di un *software*, garantirsi, all'esito dell'esecuzione del contratto, la piena ed esclusiva titolarità di tutti i diritti sul *software* oggetto di sviluppo, salvo che questo risulti eccessivamente oneroso per comprovate ragioni di carattere tecnico-economico (dal comma 2 dell'articolo 69 del CAD).

Per *software* oggetto di sviluppo, si intendono le parti di *software* effettivamente sviluppate in esecuzione del contratto; resta inteso che lo sviluppo potrebbe basarsi sull'utilizzo di componenti *software* già esistenti (es: librerie e framework open source di terzi) per le quali non è necessario acquisire titolarità ma solo licenza d'uso (che dev'essere compatibile con le finalità di riuso).

La mancata acquisizione della titolarità dell'opera non può essere utilizzata per ottenere condizioni economiche più vantaggiose, poiché non costituisce comprovata ragione di carattere tecnico-economico ai sensi dell'articolo 69 comma 2 del CAD.

Un'amministrazione, ai sensi dell'articolo 69, deve egualmente acquisire la totalità dei diritti di proprietà intellettuale e industriale su eventuali personalizzazioni o moduli *software* destinati a integrarsi o interfacciarsi con un *software* proprietario. In tal caso, l'obbligo di cui all'art. 69 avrà ad oggetto esclusivamente il modulo o la parte del *software* oggetto di sviluppo; tale modulo dovrà quindi essere separato dal resto del *software* e rilasciato secondo le modalità indicate in [3.7 Sviluppo di software ex novo](#)), avendo cura di indicare la necessaria dipendenza proprietaria nella documentazione.

Ad esempio, espressioni come quelle che seguono, ove presenti nei contratti per lo sviluppo di *software* consentono di ritenere che l'amministrazione sia titolare dei diritti nel senso richiesto dall'articolo 69 del CAD:

- «il committente sarà titolare del *software* sviluppato»;
- «la proprietà della soluzione informatica oggetto del contratto farà capo al committente o all'Amministrazione»;
- «al termine del contratto la proprietà intellettuale sulla soluzione informatica oggetto di sviluppo competerà all'amministrazione committente»;
- «tutti i diritti d'autore sul *software* sviluppato verranno trasferiti, a seguito del completamento dell'opera, all'amministrazione committente che ne diverrà titolare»;
- «tutti i diritti di sfruttamento economico sul *software* oggetto del contratto competono all'amministrazione committente».

## 1.6 Conformità del *software* alla normativa

Il riuso del *software* è un canale di amplificazione di ogni scelta in ambito informatico ed è completamente neutro rispetto alla bontà o erroneità di tali scelte. Esso può agire da moltiplicatore dell'impatto delle buone prassi o, allo stesso modo, da moltiplicatore di scelte erranee la cui diffusione non è auspicabile.

Nel promuovere il riuso e la diffusione del *software* sul quale insistono diritti di proprietà intellettuale di un'amministrazione, con un importante vantaggio economico e in termini di efficienza, è importante richiamare l'attenzione delle singole Amministrazioni sull'importanza che il *software* posto in riuso - come d'altra parte l'intero parco *software* in uso a ogni amministrazione - sia conforme alla disciplina vigente.

Poiché il processo di acquisizione di un *software* in riuso spesso comprende personalizzazioni e aggregazioni di diverse componenti, alcune delle quali potrebbero non essere più in uso o rilasciate anche anni prima, è importante ricordare che la verifica della piena conformità ai contesti normativi rimane in capo all'amministrazione che prenda in riuso un *software*, poiché solo ad essa compete la responsabilità delle decisioni assunte nell'ambito dei margini di discrezionalità assegnati e nel rispetto dei principi costituzionali di buon andamento.

## 1.7 Glossario

**Artefatti** Sottoprodotto realizzato durante lo sviluppo del *software* che aiuta a descrivere funzioni, architettura, progettazione e messa in esercizio; a solo titolo di esempio: requisiti funzionali, descrizione delle basi dati e dei processi, il set di test.

**Code Hosting (strumento di)** Una piattaforma che consente la pubblicazione di codice sorgente, organizzato in più repository. Gli strumenti di code hosting offrono spesso anche funzionalità legate all'evoluzione di un *software* quali sistemi di ticketing, processi per la contribuzione di codice da parte di terzi, area per il download dei rilasci, ecc. Nell'ambito delle presenti Linee Guida, gli strumenti scelti dalle amministrazioni devono avere dei requisiti minimi in termini di funzionalità (3.4.1 *Scelta di uno strumento di code hosting*).

**Codice sorgente** Il codice sorgente (spesso detto semplicemente «sorgente») è il testo di un programma scritto in un linguaggio di programmazione (es. C o Visual Basic) dal quale si deriva il programma finale che l'utente usa. L'accesso al codice sorgente è essenziale per poter modificare un programma.

**Community** Aggregazione di persone, fisiche e giuridiche, e risorse (ad esempio forum, chat e tecnologie per riunirsi e interagire in una località virtuale), dotata di regole e di una struttura, finalizzata alla realizzazione e/o gestione di un progetto comune.

**Formato aperto (di dato)** È un formato di dato pubblico, versionato, documentato esaustivamente e senza vincoli all'implementazione. Un formato aperto è un formato riconosciuto da un ente di standardizzazione e mantenuto in modo condiviso tra più enti che forniscono implementazioni concorrenti, con un processo trasparente. Il formato deve rimanere consistente con la versione dichiarata.

**Formato di dato** Modalità di rappresentazione del dato.

**Interoperabilità** In ambito informatico, la capacità di sistemi differenti e autonomi di cooperare e di scambiare informazioni in maniera automatica, sulla base di regole condivise.

**Licenza** In ambito informatico, il testo legale con il quale si concedono determinati diritti sul *software* e sui dati distribuiti, che altrimenti sarebbero riservati da diritti di privativa.

**Lock-in** Fenomeno di natura tecnica ed economica in cui un generico utente non riesce a svincolarsi da una scelta tecnologica precedentemente effettuata. Tale incapacità è tipicamente causata degli elevati costi legati al cambio di tecnologia ma, in molti casi, può anche dipendere dall'adozione di soluzioni proprietarie che impediscono di effettuare migrazioni. L'utilizzo di formati aperti per il salvataggio dei dati, e l'accesso libero a questi dati (soprattutto nel caso di soluzioni SaaS) sono prerequisiti per evitare fenomeni di lock-in.

**Open Source** È una modalità con cui il *software* può essere concesso in licenza. Si realizza attraverso la concessione al pubblico, dei diritti di uso, copia, modifica, distribuzione di copie anche modificate, del *software*; per fare ciò, è necessario anche che il codice sorgente sia liberamente disponibile. Altrimenti detto «software libero», «software aperto» o «software rilasciato sotto licenza aperta». L'ente certificatore delle licenze *software* corrispondenti a questa definizione è Open Source Initiative (OSI).

**Repertorio o Repository** All'interno di uno strumento di code-hosting, un repository è l'unità minima di contenimento del codice sorgente di un *software*. Il termine «repertorio» è la sua traduzione italiana (usata per esempio nel CAD Art 69, comma 1).

**Riuso** Nel contesto di queste Linee Guida, si intende il processo delineato dal CAD (art. 69) con il quale una amministrazione distribuisce («mettere a riuso») un *software* di cui ha titolarità in Open Source, a favore di altre amministrazioni che possano utilizzarlo («prendere a riuso»). Tutto il *software* a riuso è Open Source, ma non tutto il *software* Open Source è a riuso (poiché non tutto il *software* Open Source è di titolarità di una amministrazione).

**SaaS** *software as a Service*. Indica una modalità di distribuzione del *software* che non viene installato sulle postazioni degli operatori, ma che avviene attraverso l'accesso remoto a un server, per esempio collegandosi con un browser ad un indirizzo. Wikipedia, per esempio, è un *software* distribuito in modalità *software as a Service*.

**Software proprietario** *software* che ha restrizioni sul suo utilizzo, sulla sua modifica, riproduzione o redistribuzione, imposti dal titolare dei diritti di sfruttamento economico, cioè l'autore o - in caso di cessione dei diritti patrimoniali - il cessionario dei diritti in questione.

**TCO** Total Cost of Ownership: Approccio utilizzato per valutare tutti i costi del ciclo di vita di una risorsa IT calcolato su una finestra temporale adeguata al contesto della valutazione e che include il costo di migrazione verso altra soluzione (eg., acquisto, installazione, gestione, manutenzione e smantellamento). L'approccio TCO è basato sulla considerazione che il costo totale di utilizzo di risorsa IT non dipende solo dai costi di acquisto, ma anche da tutti i costi che intervengono durante l'intera vita di esercizio della risorsa stessa.

---

### Linee Guida sull'acquisizione di software

---

#### 2.1 Introduzione e contesto normativo

Per le Pubbliche amministrazioni il Codice dell'Amministrazione Digitale, di seguito denominato CAD, disciplina il riutilizzo delle soluzioni e standard aperti.

Gli articoli 68 e 69, che le presenti Linee guida hanno l'obiettivo di attuare, trattano nel medesimo contesto i temi del riutilizzo, della titolarità del software e del codice sorgente aperto per le PPAA.

Gli articoli richiamati sono stati modificati sia dal d.lgs. 26 agosto 2016 n. 179 sia dal d.lgs. 12 gennaio 2017 n. 217. L'ultimo aggiornamento ha comportato:

- la modifica dell'art. 68, abrogando il comma 3 e trasponendo il contenuto nell'art. 1 co. 3, lettere l-bis e l-ter;
- la riformulazione dell'art. 69, comma 2;
- l'introduzione del comma 2 bis dell'art. 69;
- l'abrogazione dell'articolo 70 rubricato banca dati dei programmi informatici riutilizzabili.

Il testo dell'art. 68 è rimasto immutato, eccezion fatta per l'aggiornamento del riferimento normativo al D.Lgs. 50/2016<sup>1</sup> in luogo del richiamo alla precedente normativa in materia di appalti.

Fino alla modifica apportata dal D.Lgs. 217/2017, nell'acquisizione di software da parte delle pubbliche amministrazioni svolgevano un ruolo:

- il mercato elettronico: «*strumento di acquisto e di negoziazione che consente acquisti telematici per importi inferiori alla soglia di rilievo europeo basati su un sistema che attua procedure di scelta del contraente interamente gestite per via telematica*» (di particolare rilevanza il MePA gestito da CONSIP);
- le convenzioni quadro e gli accordi quadro stipulati, ai sensi della normativa vigente, da CONSIP e dai soggetti aggregatori;
- il Catalogo nazionale programmi riutilizzabili gestito dall'AgID.

I primi due continuano a svolgere la funzione, mentre le funzioni del catalogo, abrogato in quanto tale dal CAD, sono assunte dal portale *Developers Italia* (<https://developers.italia.it>), che assume il ruolo di «piattaforma», *rectius repertorio* - secondo la dizione dell'art. 69 co. 1 -, e delle *piattaforme* di cui all'art. 69 co. 2bis.

---

<sup>1</sup> <http://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.legislativo:2016-04-18;50!vig=>

Il presente documento ribadisce che i «principi di economicità e di efficienza, tutela degli investimenti, riuso e neutralità tecnologica» (comma 1 dell'art. 68 del CAD<sup>2</sup>) si raggiungono attuando quanto previsto dal comma 2 dell'art. 69 del CAD<sup>3</sup>: «il riuso dei programmi informatici di proprietà delle pubbliche amministrazioni» garantendo che queste ultime, oltre ad essere titolari del software, rendano il software Open Source attraverso l'apposizione di una licenza aperta.

## 2.2 Oggetto della valutazione

La valutazione comparativa deve essere svolta quando le pubbliche amministrazioni intendano acquisire «*programmi informatici o parti di essi*». L'oggetto della valutazione quindi è un *software* (come identificato in 1.2. \**software* oggetto di queste linee guida\*) che risponda a specifiche esigenze funzionali dell'amministrazione.

A titolo esemplificativo, rimane all'esterno del perimetro di questo documento l'acquisizione di sole componenti hardware dei sistemi informativi (server, postazioni di lavoro, stampanti, ecc.).

Ulteriori situazioni dove non è applicabile il percorso decisionale proposto in questo Capitolo 2 possono riguardare ad esempio:

- l'adesione a contratti quadro già sottoscritti, nel caso in cui tale adesione abbia un carattere obbligatorio per le amministrazioni;
- accordi quadro, in quanto strumenti che definiscono esclusivamente le clausole generali che, in un determinato periodo temporale, regolano i contratti da stipulare (le caratteristiche specifiche della singola fornitura vengono successivamente definite in appositi Appalti Specifici);
- completamento di progetti o realizzazioni per le quali la valutazione comparativa sia già stata effettuata preliminarmente all'acquisizione iniziale;
- gare che abbiano come oggetto l'outsourcing completo dei sistemi informativi, in quanto la scelta dell'esternalizzazione riguarda un ambito strategico che esula dallo specifico contesto delle presenti Linee guida e risponde a scelte di governance dell'amministrazione e a obiettivi di carattere strategico di ordine più generale.

Si noti che nei casi qui elencati si devono comunque applicare le [Linee Guida sul riuso del Software](#) descritte nel Capitolo 3.

## 2.3 Valutazione comparativa

### 2.3.1 Descrizione delle soluzioni

L'articolo 68, comma 1 del CAD indica le tipologie di soluzione oggetto dell'analisi comparativa per tipologia di software:

«Le pubbliche amministrazioni acquisiscono programmi informatici o parti di essi nel rispetto dei principi di economicità e di efficienza, tutela degli investimenti, riuso e neutralità tecnologica, a seguito di una valutazione comparativa di tipo tecnico ed economico tra le seguenti soluzioni disponibili sul mercato:

- a) *software* sviluppato per conto della pubblica amministrazione;
- b) riutilizzo di *software* o parti di esso sviluppati per conto della pubblica amministrazione;
- c) *software* libero o a codice sorgente aperto;
- d) *software* fruibile in modalità cloud computing;
- e) *software* di tipo proprietario mediante ricorso a licenza d'uso;
- f) *software* combinazione delle precedenti soluzioni.»

---

<sup>2</sup> <http://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.legislativo:2005-03-07;82!vig=~art68>

<sup>3</sup> <http://www.normattiva.it/uri-res/N2Ls?urn:nir:stato:decreto.legislativo:2005-03-07;82!vig=~art69>

La seguente lista di definizioni descrive le sei soluzioni previste dalla normativa:

- A - software sviluppato per conto della pubblica amministrazione** Soluzione detta anche «opzione *make*»: la PA affida lo sviluppo del *software* (sia esso ex novo o modifica di *software* esistente) a un fornitore e quest'ultimo si impegna a consegnare alla P.A. il software sviluppato sulla base dei requisiti da questa definiti. Per esempio, nel ciclo di vita del *software* (analisi, progettazione, sviluppo, collaudo, rilascio, manutenzione) la P.A. potrebbe occuparsi delle fasi di analisi e progettazione, definendo i requisiti del software, per poi affidare lo sviluppo al fornitore.
- B - Riutilizzo di software o parti di esso sviluppati per conto della pubblica amministrazione** Soluzione «riuso» di un *software* della P.A. (o suoi componenti) già esistente e disponibile.
- C - software libero o a codice sorgente aperto** *software* con licenza Open Source (vedi [1.7 Glossario](#)). In particolare, si intende tutto il *software* distribuito sotto una licenza certificata da OSI ([lista completa](#)), come descritto in [3.5.2 Licenze per il software aperto](#).
- D - software fruibile in modalità cloud computing** Soluzione nella quale la P.A. acquisisce il *software* come servizio. In questa soluzione non sono ricomprese le soluzioni HaaS (*Hardware as a Service*) e IaaS (*Infrastructure as a Service*).
- E - software di tipo proprietario mediante ricorso a licenza d'uso** *software* soggetto a condizioni di licenza d'uso di tipo proprietario da installare «*on premise*».
- F - software combinazione delle precedenti soluzioni** *software* realizzato con componenti appartenenti a più di una categoria tra quelle precedenti. Ad esempio, *software* in cui una soluzione in riuso si appoggia su un *middleware Open Source* e accede a un database proprietario, con componenti realizzate appositamente per conto dell'amministrazione destinataria della soluzione. È di fatto la tipologia più comune tra quelle effettivamente in uso nelle pubbliche amministrazioni.

Inoltre, il comma 1 dell'articolo 69 del CAD indica che

»le pubbliche amministrazioni che siano titolari di soluzioni e programmi informatici realizzati su specifiche indicazioni del committente pubblico, hanno l'obbligo di rendere disponibile il relativo codice sorgente, completo della documentazione e rilasciato in repertorio pubblico sotto licenza aperta, in uso gratuito.»

Per quanto riguarda la soluzione D) (Software fruibile in modalità *cloud computing*), questa è da considerarsi una modalità di erogazione di servizi informatici, quindi ascrivibile di volta in volta nelle categorie A), B), C) ed E).

### 2.3.2 Descrizione dei criteri per la valutazione

Di seguito si riporta una breve descrizione dei criteri necessari per la valutazione comparativa tra le soluzioni, per ognuno dei criteri elencati al comma 1-bis dell'art. 68 del CAD.

**Costo complessivo** Nell'ambito del presente documento è da intendersi come *Total Cost of Ownership* (TCO) della soluzione, calcolato su una finestra temporale adeguata al contesto della valutazione, comprensivo del costo di migrazione verso altra soluzione (vedere anche paragrafo 2.4).

**Utilizzo di formati di dati aperti** Uso, da parte della soluzione da valutare, di formati standard e aperti (v. [Glossario](#)) per la rappresentazione di dati, metadati e documenti, finalizzata ad assicurare l'interoperabilità tra i sistemi informatici delle pubbliche amministrazioni e/o dei gestori di pubblici servizi.

**Utilizzo di interfacce aperte** Uso, da parte della soluzione da valutare, di interfacce aperte, incluse *Application Programming Interface* (API), vale a dire interfacce pubbliche, documentate e liberamente implementabili/estendibili, finalizzata ad assicurare l'interoperabilità tra i sistemi informatici delle pubbliche amministrazioni e/o dei gestori di pubblici servizi.

**Utilizzo di standard per l'interoperabilità** Adeguatezza della soluzione da valutare ad assicurare l'interoperabilità tra i sistemi informatici delle pubbliche amministrazioni e/o dei gestori di pubblici servizi.

**Livelli di sicurezza** Nell'ambito del presente documento è da intendersi come l'esistenza di adeguate garanzie in merito ai livelli di sicurezza della soluzione indipendentemente dalla natura giuridica del titolare del software e/o dell'erogatore del servizio in modalità *cloud computing*.

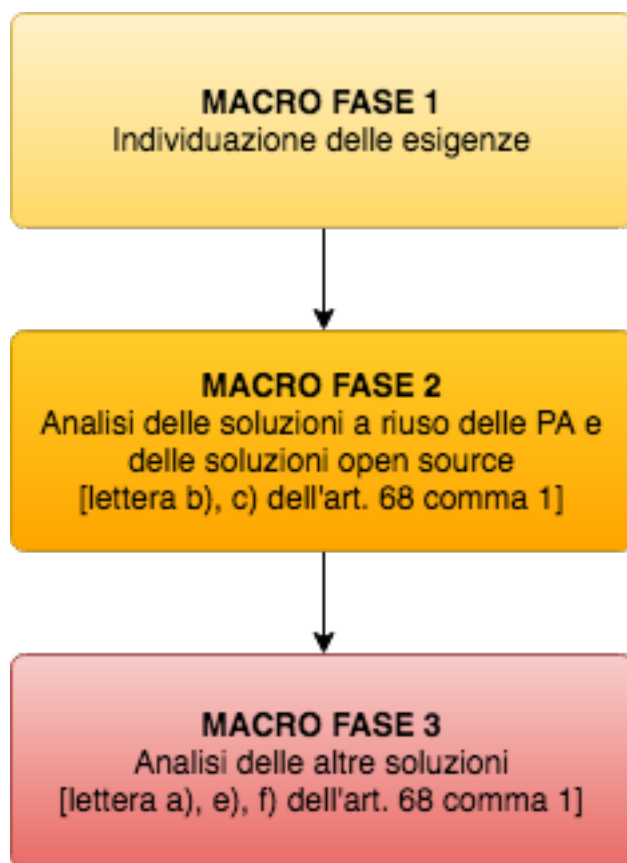
**Conformità alla normativa in materia di protezione dei dati personali** È da intendersi come la conformità dei processi/procedure alla normativa in materia di protezione dei dati personali, indipendentemente dalla natura giuridica del titolare del software e/o dell'erogatore del servizio in modalità *cloud computing*.

**Livelli di servizio del fornitore** È da intendersi come la capacità del fornitore di erogare i servizi nel rispetto delle metriche precedentemente individuate dalla pubblica amministrazione in un *Service Level Agreement* (SLA).

### 2.3.3 Descrizione delle macro-fasi

Vista l'eterogeneità delle soluzioni e la difficoltà ad effettuare comparazioni quantitative omogenee, come in caso di confronto tra una soluzione dalla quale possano essere ricavati costi certi (soluzione proprietaria in modalità *on premise* o in modalità *cloud computing*) e una soluzione da realizzare *ex novo* - per la quale si disponga soltanto dello studio di fattibilità - si è preferito indicare un processo decisionale attraverso la descrizione di Fasi e la loro organizzazione in Macro fasi.

La seguente immagine riporta le Macro fasi che caratterizzano il processo decisionale per dare seguito alla valutazione comparativa prevista all'articolo 68 del CAD.



Le Macro fasi individuate sono:

**MACRO FASE 1:** Ha l'obiettivo di definire le esigenze specificando i bisogni e i vincoli (organizzativi ed economici) che condizionano le scelte per l'identificazione di una soluzione adeguata alle esigenze dell'amministrazione;

**MACRO FASE 2:** Qui la pubblica amministrazione accerta la possibilità di soddisfare le proprie esigenze utilizzando una soluzione già in uso presso altre amministrazioni (di seguito «soluzioni a riuso delle PA») o a *software* libero o codice sorgente aperto (di seguito «soluzioni Open Source»).

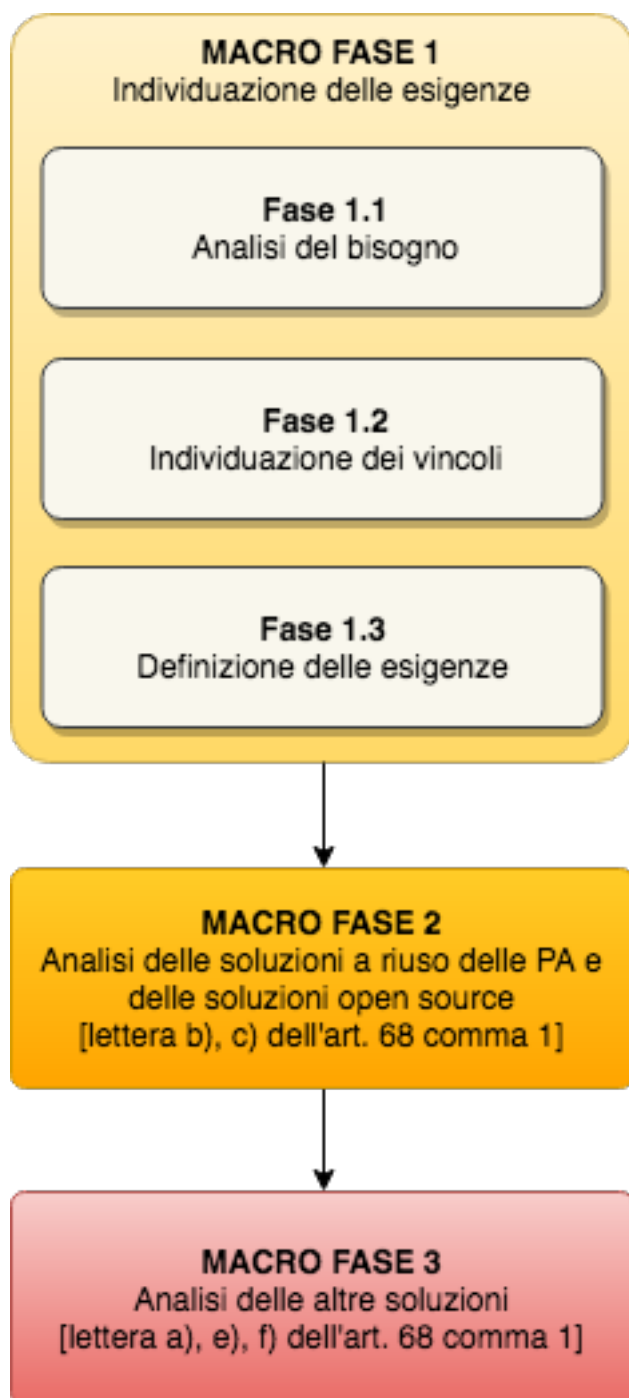


**MACRO FASE 3:** Ove la Macro fase 2 non permetta di rispondere alle esigenze della Pubblica amministrazione, si persegue il soddisfacimento delle stesse attraverso il ricorso a programmi informatici di tipo proprietario, mediante ricorso a licenza d'uso e/o a realizzazioni ex novo.

In quanto segue le Macro fasi individuate sono suddivise in Fasi, descrivendo le attività da realizzare in termini di criteri e metodologie da adottare.

## 2.4 Macro fase 1: Individuazione delle esigenze

Nella presente Macro fase, la pubblica amministrazione definisce le esigenze (bisogni e vincoli) che condizionano le scelte per l'individuazione di una soluzione; si suggerisce di redigere un documento (senza alcun vincolo di forma) descrittivo delle esigenze individuate da utilizzare nelle fasi che riguardano la comparazione e valutazione delle soluzioni.



### 2.4.1 Fase 1.1: Analisi del fabbisogno

L'amministrazione definisce il proprio fabbisogno, per l'individuazione della soluzione software. Nel fare questo tiene conto di quanto definito nel Programma degli acquisti e nella Programmazione dei lavori pubblici (art. 21 del D. Lgs. n. 50 del 18/04/2016).

Le attività previste nella presente fase sono:

- studio del contesto attraverso la descrizione delle caratteristiche dell'amministrazione: finalità, struttura ed organizzazione;
- descrizione dei flussi operativi interessati dal *software* da acquisire, che la pubblica amministrazione mette in atto per dare seguito alle procedure amministrative;
- ipotesi di ottimizzazione dei flussi in relazione al *software* da acquisire;
- identificazione degli «strumenti» (definizione degli obiettivi) necessari alla realizzazione dei processi operativi individuati;
- enunciazione dei requisiti, cioè dei bisogni a cui il software deve rispondere, prevedendo una differenziazione tra requisiti indispensabili e non.

La presente fase si conclude con la:

- individuazione dei bisogni della pubblica amministrazione.

### 2.4.2 Fase 1.2: Individuazione dei vincoli

L'amministrazione descrive i vincoli che condizionano l'approvvigionamento della soluzione software.

Le attività previste in questa fase sono:

- individuazione della **disponibilità di bilancio** (di seguito  $T_{\text{bilancio}}$ ) per assicurare la disponibilità e la messa in produzione della soluzione da acquisire (eventuale bonifica dati e migrazione da sistemi preesistenti, installazione, personalizzazione, integrazione con sistemi esistenti, formazione, supporto all'avvio, attività gestionali, pagamento di eventuali eccedenze, ecc.);
- stima dei **tempi per la messa in esercizio** (di seguito  $T_{\text{tempi}}$ ) della soluzione che la Pubblica amministrazione può assorbire;
- eventuali norme e linee guida che il *software* dovrà seguire nella propria implementazione tecnica, come per esempio:
  - [Linee Guida AgID sul Design](#) (che includono anche l'adempimento alla Legge 4/2004 sull'accessibilità);
  - [Linee Guida AgID sull'Interoperabilità](#);
  - [Linee Guida AgID sullo sviluppo di \\*software\\* sicuro](#).
- eventuali altri vincoli di interesse dell'amministrazione.

La presente fase si conclude con la:

- individuazione dei vincoli (economici e temporali) che condizionano le scelte dell'amministrazione.

### 2.4.3 Fase 1.3: Redazione del documento descrittivo delle esigenze

L'amministrazione redige il documento descrittivo delle esigenze da utilizzarsi nelle fasi successive della valutazione comparativa.

Le attività previste in questa fase sono:

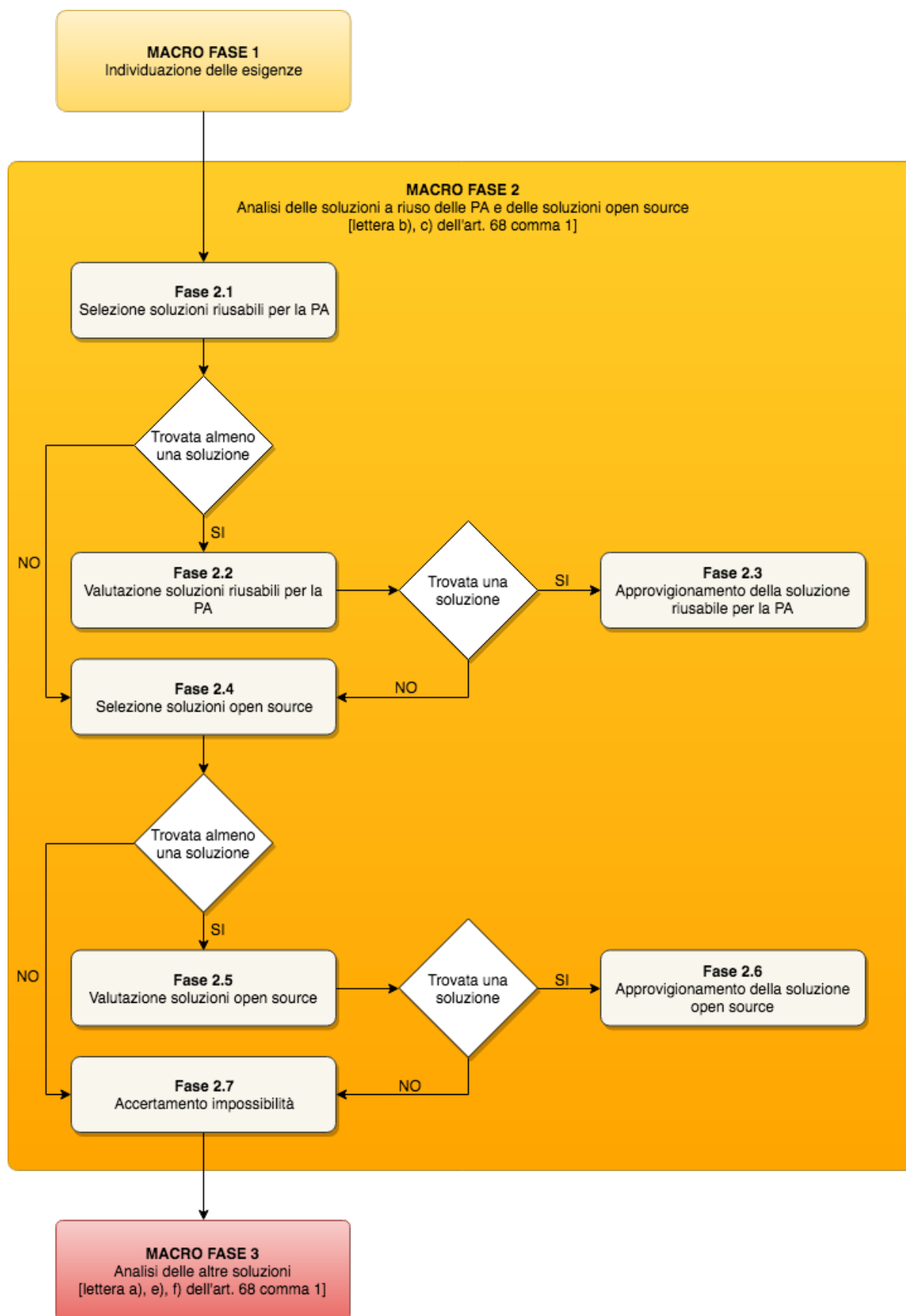
- redazione del documento descrittivo delle esigenze che contiene le evidenze delle precedenti fasi 1.1 e 1.2.

La presente fase si conclude con la:

- disponibilità del documento descrittivo delle esigenze.

## **2.5 Macro fase 2: Analisi delle soluzioni a riuso delle PA e delle soluzioni Open Source**

La pubblica amministrazione, a partire dalla disponibilità di «soluzioni a riuso delle PA» e «soluzioni Open Source», deve verificare il soddisfacimento delle proprie esigenze in tali soluzioni.



Al fine di razionalizzare la spesa complessiva delle pubbliche amministrazioni la verifica di soddisfacimento delle esigenze deve prima considerare le «soluzioni a riuso delle PA» e successivamente le «soluzioni Open Source».

L'attuazione dell'art. 69 del CAD assicura che le «soluzioni a riuso delle PA» rendano disponibile il relativo codice sorgente, completo della documentazione, in repertorio pubblico sotto licenza aperta.

Per approfondimenti in merito si rimanda a *3 Linee Guida sul riuso del software (art. 69)*.

### 2.5.1 Fase 2.1: Ricerca soluzioni riusabili per la PA

L'amministrazione individua le «soluzioni a riuso delle PA» candidate a soddisfare le proprie esigenze.

Le attività previste in questa fase sono:

- ricerca delle «soluzioni a riuso delle P.P.A.A.» presenti all'interno della piattaforma Developers Italia.

La presente fase si conclude con la:

- individuazione delle «soluzioni a riuso delle P.P.A.A.» di interesse per la Pubblica amministrazione.

### 2.5.2 Fase 2.2: Valutazione soluzioni riusabili per la PA

Se la precedente fase 2.1 ha permesso di individuare almeno una delle «soluzioni a riuso delle PA» potenzialmente di interesse per la Pubblica amministrazione, attraverso la realizzazione della presente fase di valutazione si provvede ad individuare la «soluzione a riuso della PA» migliore. Sono eleggibili in questa fase anche soluzioni che soddisfano la maggior parte dei requisiti e hanno bisogno di modifiche o personalizzazioni.

Per ognuna delle «soluzioni a riuso delle PA» potenzialmente d'interesse si provvede a:

- verificare in prima battuta la conformità almeno dichiarata alle normative vigenti, come rinvenibile nella scheda del software presente su Developers Italia. In particolare:
  - la conformità alle regole sull'interoperabilità prescritte dalle linee guida emanate in attuazione dell'articolo 73 del CAD;
  - la conformità alle normative sulla protezione dei dati personali;
  - la conformità ai livelli minimi di sicurezza previsti per le pubbliche amministrazioni.
  - la conformità ai requisiti di accessibilità (Legge 4/2004).
- valutare la qualità della soluzione attraverso i seguenti parametri, alcuni dei quali possono essere rinvenuti nella scheda del software presente su Developers Italia:
  - grado di copertura dei requisiti, funzionali e non funzionali;
  - presenza di un manutentore del *software* in questione;
  - eventuale presenza di accordi con terzi stipulati dall'amministrazione titolare e utilizzabili dall'amministrazione valutante, riguardo attività di supporto per l'installazione e/o la personalizzazione della soluzione o comunque le modalità di fruibilità della soluzione stessa (es: una in-house regionale può mettere a riuso *software* Open Source per i propri comuni assieme ad un accordo di fornitura di servizi di installazione e formazione);
  - presenza di vincoli e dipendenze obbligatorie con altro software aperto e/o con *software* proprietario; per esempio, un software Open Source potrebbe richiedere necessariamente una licenza per un database proprietario, oppure potrebbe necessitare una licenza per una API proprietaria di un servizio *cloud*;
  - presenza e grado di competenza delle risorse interne alla PA in merito alla gestione degli ambienti e dei linguaggi utilizzati nella soluzione;
  - numero e tipologia di altre pubbliche amministrazioni che utilizzano il progetto open source;
  - vitalità del progetto open source, attraverso la valutazione di indicatori visibili sul repository, quali per esempio frequenza delle modifiche (*code activity*), frequenza dei rilasci (*release history*), comunità degli utenti (*user community*), longevità del progetto (*longevity*), numero sviluppatori unici.

- stimare il Total Cost of Ownership ( $S_{TCO}$ ) come descritto in *2.7 Total Cost of Ownership (TCO)*, con particolare attenzione a:
  - eventuali costi di installazione del *software* nel Cloud della PA oppure costi per fruizione del *software* tramite modalità SaaS ove presente nel Marketplace Cloud di AgID;
  - eventuali costi per la formazione del personale, considerando sia quelli necessari per l'addestramento dei soggetti destinati alla gestione della soluzione sia quelli per il suo utilizzo da parte degli utenti finali;
  - eventuali costi necessari all'integrazione della soluzione con i propri sistemi;
  - eventuali costi di personalizzazione, compresi quelli necessari ad assicurare dei requisiti funzionali e non funzionali, non già presenti nel *software* a riuso;
  - eventuali costi per la verifica di compliance alle normative vigenti.
- stimare i tempi per la messa in produzione della soluzione (di seguito  $S_{tempi}$ );
- eventuali altri stime espressione della specificità dell'amministrazione.

Per dare seguito alla valutazione di quanto indicato nei precedenti punti la pubblica amministrazione dovrà provvedere, ove non già in suo possesso, a recuperare tutte le necessarie informazioni nelle modalità previste dalla norma.

Pertanto se:

- Il costo (TCO) rientri nei vincoli di bilancio stabiliti ( $S_{TCO} < T_{bilancio}$ );
- I tempi di messa in produzione sono compatibili con i tempi stimati ( $S_{tempi} < T_{tempi}$ );
- Sono rispettati eventuali altri vincoli ostativi.

l'amministrazione individua la soluzione più rispondente alle proprie esigenze in base alla valutazione effettuata.

La presente fase si conclude con la:

- determinazione della migliore «soluzione a riuso delle P.P.A.A.», o eventuale non presenza di una soluzione.

### 2.5.3 Fase 2.3: Approvvigionamento della soluzione riusabile per la PA

Ove a seguito della precedente fase 2.2 l'amministrazione abbia determinato una «soluzione a riuso della PA» che soddisfa le sue esigenze, provvede all'approvvigionamento. Il processo di riuso è descritto in *3.9 Riuso di un software o utilizzo di un software Open Source*.

**La valutazione comparativa si considera conclusa.**

Nel caso in cui la Pubblica Amministrazione debba sostenere dei costi durante la fase di approvvigionamento (es: personalizzazione, installazione, formazione), la stessa acquisisce tali servizi così come previsto dal D.Lgs. 50/2016 s.m.i. (di seguito *Codice dei contratti pubblici*).

### 2.5.4 Fase 2.4: Ricerca soluzioni Open Source

Se non è possibile individuare una «soluzione a riuso della PA», l'amministrazione DEVE ampliare la ricerca delle soluzioni da considerare per il soddisfacimento delle proprie esigenze alle «soluzioni Open Source», cioè *software* rilasciato sotto licenza aperta ma non di titolarità di una Pubblica Amministrazione e quindi non pubblicato a riuso. Sono eleggibili in questa fase anche soluzioni che soddisfano la maggior parte dei requisiti e hanno bisogno di modifiche o personalizzazioni.

Le attività previste nella presente fase sono:

- ricerca di progetti di *software* Open Source la cui titolarità è attribuita a soggetti diversi dalle pubbliche amministrazioni. Tale ricerca deve almeno essere effettuata con gli strumenti messi a disposizione su Developers Italia, e può anche essere estesa su altre piattaforme internazionali che gestiscono progetti di *software* Open Source.

La ricerca effettuata dalla pubblica amministrazione deve verificare:

- se la licenza sotto la quale il *software* è tra quelle suggerite nel presente documento o certificata da OSI ([lista completa](#)).
- se la licenza è compatibile con le licenze di *software* con cui la si vuole integrare e/o con l'uso che se ne vuole fare;

La presente fase si conclude con la:

- individuazione delle «soluzioni Open Source» di interesse per la pubblica amministrazione

### 2.5.5 Fase 2.5: Valutazione soluzioni Open Source

Se la precedente fase 2.4 ha permesso di individuare almeno una delle «soluzioni Open Source» potenzialmente di interesse, attraverso la fase di valutazione si individua la «soluzione Open Source» migliore per le esigenze della Pubblica Amministrazione.

La valutazione del *software* open source in questa fase deve seguire gli stessi criteri di valutazione descritti per la fase 2.2. Si consideri quindi la fase 2.5 come un doppiante della fase 2.2, applicata ad un diverso insieme di *software* (open source di terzi invece che *software* a riuso).

La presente fase si conclude con la:

- determinazione della migliore «soluzione Open Source», o eventuale non presenza di una soluzione.

### 2.5.6 Fase 2.6: Approvvigionamento della soluzione Open Source

Ove a seguito della precedente fase 2.5 l'amministrazione abbia individuato una «soluzione Open Source» che soddisfa le sue esigenze, procede all'acquisizione. Il processo di acquisizione è descritto in [3.9 Riuso di un software o utilizzo di un software Open Source](#).

**La valutazione comparativa si considera conclusa.**

Nel caso in cui la Pubblica Amministrazione debba sostenere dei costi durante la fase di approvvigionamento (es: personalizzazione, installazione, formazione), la stessa acquisisce tali servizi così come previsto dal *Codice dei contratti pubblici*.

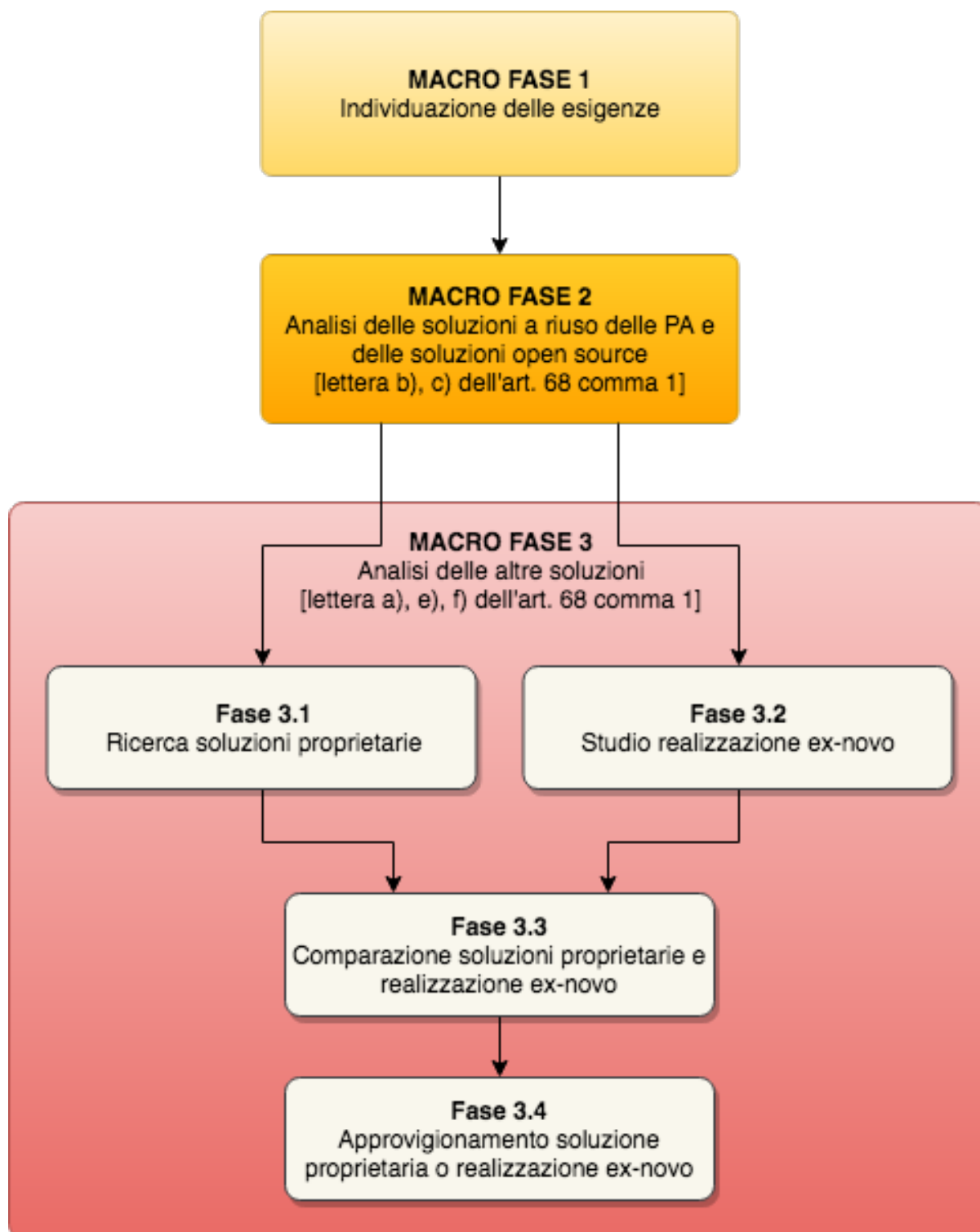
### 2.5.7 Fase 2.7: Accertamento impossibilità

Nel caso in cui sia accertata l'impossibilità di individuare una soluzione che soddisfi almeno in larga misura le esigenze dell'amministrazione tra le «soluzioni a riuso della PA» e le «soluzioni Open Source», si procede alla redazione di un documento (senza vincoli di forma) che motivi le ragioni dell'accertata impossibilità, da conservare agli atti del procedimento.

La pubblica amministrazione prosegue la valutazione comparativa dando seguito alle Fasi previste nella successiva Macro fase 3.

## 2.6 Macro fase 3: Analisi delle altre soluzioni

La pubblica amministrazione per soddisfare le proprie esigenze DEVE esaminare contestualmente le opportunità offerte dalle soluzioni proprietarie e quelle di una realizzazione ex novo.



### 2.6.1 Fase 3.1: Ricerca soluzioni proprietarie

La pubblica amministrazione deve valutare le soluzioni proprietarie presenti sul mercato.

L'amministrazione deve effettuare la ricerca di una soluzione con licenza d'uso proprietaria, analizzando le offerte secondo quanto indicato dal Codice dei contratti pubblici.

L'amministrazione deve verificare che il *software* in licenza rispetti i seguenti vincoli (ossia, **la mancanza anche di uno solo di questi rende la soluzione non eleggibile**):



- conformità alle regole sull'interoperabilità prescritte dalle linee guida emanate in attuazione dell'articolo 73 del CAD;
- conformità alle normative sulla protezione dei dati personali;
- conformità ai livelli minimi di sicurezza previsti per le pubbliche amministrazioni;
- la conformità ai requisiti di accessibilità (Legge 4/2004);
- possibilità di esportare gratuitamente, in ogni momento, l'intera base di dati (inclusi di ogni tipo di indice o metadato utilizzato per implementare le funzionalità del *software* stesso) in formato standard, aperto e documentato, per scongiurare la possibilità di lock-in, come meglio specificato nelle [linee guida n.8 di ANAC](#).

Tra i *software* che rispettano i vincoli appena indicati, l'amministrazione effettua una analisi comparativa che tenga conto dei seguenti criteri:

- assicurare la soddisfazione dei requisiti funzionali e non determinati nella Macro Fase 1 con quelli indicati nella documentazione;
- verificare l'idoneità della soluzione ad interoperare con i sistemi già in uso presso l'amministrazione;
- eventuali costi di installazione del *software* nel Cloud della PA oppure costi per fruizione del *software* tramite modalità SaaS ove presente nel Marketplace Cloud di AgID;
- eventuali costi necessari all'integrazione della soluzione con i sistemi già in uso presso l'amministrazione;
- eventuali costi per la formazione del personale destinato alla gestione e amministrazione della soluzione esaminata;
- calcolo del TCO e sua congruità rispetto alla disponibilità di bilancio determinata nella precedente Macro fase 1.

La presente fase si conclude con la:

- individuazione delle soluzioni con licenza d'uso proprietaria che soddisfano le esigenze dell'amministrazione.

### 2.6.2 Fase 3.2: Studio realizzazione ex novo

La Pubblica amministrazione, dopo aver individuato l'esistenza o meno di una soluzione proprietaria confacente ai propri bisogni, elabora un documento contenente un progetto di fattibilità<sup>1</sup> contenente la stima delle attività, dei costi e dei tempi da sostenere per la realizzazione di una soluzione ex novo che soddisfi completamente le esigenze indicate nel documento sull'analisi dei fabbisogni così come descritto in [2.4.1 Fase 1.1: Analisi del fabbisogno](#).

### 2.6.3 Fase 3.3: Comparazione soluzioni proprietarie e realizzazione ex novo

Nella valutazione tra lo sviluppo di una soluzione ex novo e l'acquisto di una proprietaria (cosiddetta valutazione «make or buy»), l'amministrazione verifica vantaggi e svantaggi di entrambe le soluzioni, tenendo come riferimento il seguente elenco:

#### **Vantaggio acquisto soluzione proprietaria:**

- rapida messa in esercizio;
- garanzia totale e rischio applicativo a carico del fornitore;
- manutenzione del fornitore;
- costi di acquisto o abbonamento più bassi rispetto ad uno sviluppo completo;

#### **Vantaggio sviluppo soluzione ex novo:**

- esaustività dei propri fabbisogni e obiettivi;

---

<sup>1</sup> Analisi di Fattibilità per l'acquisizione delle forniture ICT

- facilità di gestione (importazione ed esportazione) dei propri dati;
- TCO (Total Cost of Ownership) a medio/lungo termine;
- condivisione della soluzione e quindi ottimizzazione sui costi della sua tenuta in vita;
- estensione e aggiornamento;
- *riuso da parte di altre amministrazioni.*

**Svantaggi di una soluzione proprietaria:**

- licenze periodiche (abbonamenti mensili, annuali) o aggiornamenti di versione a pagamento;
- rigidità del flusso operativo o meglio non adattabile all'organizzazione operativa della Pubblica Amministrazione;
- possibilità di lock-in, cioè costi eccessivi di cambio di soluzione in futuro;
- stabilità economica del fornitore.

**Svantaggi di una soluzione sviluppata ex novo:**

- maggiori attività da eseguire;
- maggiore necessità di coordinamento;
- tempi di messa in produzione maggiori.

## **2.6.4 Fase 3.4: Approvvigionamento soluzione proprietaria o realizzazione ex novo**

A seguito della precedente fase 3.3 l'amministrazione ha determinato una soluzione, con licenza proprietaria o da realizzarsi ex novo, che soddisfa le sue esigenze e provvede all'approvvigionamento della stessa secondo le procedure previste dal Codice dei contratti pubblici.

Nel caso in cui si opti per la realizzazione ex novo, considerando i commi 1 e 2 dell'Articolo 69 che disciplinano la messa a riuso del software che verrà realizzato, si rimanda a *3.7 Sviluppo di software ex novo* per le informazioni su come progettare questa realizzazione per adempiere ai commi citati e metterlo così a riuso.

Nel caso che si proceda ad una acquisizione di *software* proprietario sotto licenza, si ricorda che l'Amministrazione deve ove possibile acquisire la titolarità del codice sviluppato (come spiegato in *1.5 Titolarità*), per metterlo a riuso.

**La valutazione comparativa si considera conclusa.**

## **2.7 Total Cost of Ownership (TCO)**

Le valutazioni comparative economiche sono effettuate attraverso l'utilizzo di strumenti economico/finanziari quali il TCO (Total Cost of Ownership), che rappresenta il costo globale di un bene durante il suo ciclo di vita. Il TCO prende in considerazione sia i costi diretti che i costi indiretti, rappresentando il metodo consigliato sia per misurare i costi totali (attraverso l'identificazione di tutte le spese, in termini chiari e facilmente misurabili), sia per effettuare la suddetta valutazione comparativa durante la fase 3.

Per eseguire correttamente un TCO, è necessario calcolare i costi dell'intero ciclo di vita del *software* e non solo quelli meramente necessari alla sua acquisizione, come per esempio:

- Costi per l'acquisto di hardware e licenze *software* proprietarie necessari alla messa in esercizio del *software* che si sta acquisendo;
- Costi per la personalizzazione del software;
- Costi di manutenzione (correttiva e evolutiva);
- Costi di formazione del personale;

- Costi di migrazione dei dati da precedenti soluzioni.

In questa linea guida, è stato indicato di utilizzare il TCO in due diverse fasi: durante la Macro Fase 2 (come strumento per stimare il costo relativo all'acquisizione di un *software* open source), e durante la Macro Fase 3 (come strumento per valutare una eventuale scelta tra la realizzazione ex-novo e l'acquisizione di software proprietario).

Ai fini dell'adempimento a questa Linea Guida, quindi, si richiede che in fase di analisi comparativa l'amministrazione effettui una stima dei costi secondo il modello qui delineato, tenendo sempre conto di tutto il ciclo di vita. Questa analisi dovrà essere tanto più accurata quanto grossa è l'acquisizione necessaria.

Fornire modelli di calcolo del TCO per le esigenze delle amministrazioni va oltre gli obiettivi di questa Linea Guida, poiché le esigenze sono le più differenti. AgID potrà, in futuro, pubblicare dei modelli facoltativi, pronti all'uso, all'interno della piattaforma Developers Italia.

## 2.8 Scelta della modalità di erogazione del software

Nel caso in cui il *software* (a riuso, open source o proprietario) debba essere installato su server, l'amministrazione si potrebbe trovare a valutare la modalità di erogazione tra le seguenti opzioni:

1. utilizzo in modalità SaaS, qualora il *software* sia disponibile sotto forma di servizio SaaS nel Marketplace Cloud e dunque sia qualificato ai sensi della circolare AgID «*Criteri per la qualificazione di servizi SaaS per il Cloud della PA*»;
2. installazione su un server nella disponibilità diretta dell'amministrazione.

La scelta tra queste opzioni deve avvenire tramite il calcolo del Total Cost of Ownership come descritto nella sezione 2.7. *Total Cost of Ownership (TCO)*.

Secondo quanto disciplinato nel capitolo 3 del Piano Triennale per l'Informatica nella PA, ai fini dell'installazione su server nella propria disponibilità l'Amministrazione deve ricorrere all'utilizzo del cosiddetto Cloud della PA, scegliendo una delle seguenti opzioni di tipo IaaS:

- installazione in PSN (Poli Strategici Nazionali);
- installazione in Cloud SPC Lotto 1;
- installazione in Cloud Service Provider Qualificati ai sensi della circolare AGID «*Criteri per la qualificazione dei Cloud Service Provider per la PA*».



---

## Linee Guida sul riuso del *software* (art. 69)

---

### 3.1 Introduzione e contesto normativo

Il comma 1 dell'art. 69 definisce l'**obbligo**, per le pubbliche amministrazioni titolari di *software* realizzato su specifiche indicazioni del committente pubblico, «di rendere disponibile il relativo codice sorgente, completo della documentazione e rilasciato in repertorio pubblico sotto licenza aperta, in uso gratuito ad altre pubbliche amministrazioni o ai soggetti giuridici che intendano adattarli alle proprie esigenze».

La nuova formulazione dell'art. 69, ai commi 2 e 2 bis, riportati di seguito, sottolinea lo scopo di favorire il riuso disponendo «che l'amministrazione committente sia sempre titolare di tutti i diritti sui programmi e i servizi delle tecnologie dell'informazione e della comunicazione appositamente sviluppati per essa», «salvo che ciò risulti eccessivamente oneroso per comprovate ragioni di carattere tecnico-economico» e che «il codice sorgente, la documentazione e la relativa descrizione tecnico funzionale di tutte le soluzioni informatiche... sono pubblicati attraverso una o più piattaforme individuate dall'AgID con proprie linee guida da adottarsi ai sensi dell'articolo 71».

### 3.2 Modello di riuso

Si descrive nel dettaglio il modello di riuso delineato dal CAD. Ciascun punto del seguente flusso viene specificato in una sezione successiva del presente documento.

#### Fase di sviluppo

1. L'amministrazione «A», effettuando la valutazione comparativa prevista dall'art. 68, secondo quanto descritto in ref:*acquisizione*, decide che, per soddisfare le proprie esigenze, deve necessariamente ricorrere alla realizzazione integrale di un *software* ex novo o la personalizzazione di un *software* Open Source esistente
2. L'amministrazione «A» utilizza proprie risorse e/o ricorre ad un appalto per realizzare il software. In caso di appalto, come richiesto dall'art. 69 comma 2, l'amministrazione si garantisce l'acquisizione della titolarità di tutti i diritti di proprietà intellettuale e industriale sul *software* commissionato (1.5 *Titolarità*).
3. Durante il corso della realizzazione del *software* e/o al termine della stessa, l'amministrazione pubblica il codice sorgente del proprio *software* sotto una **licenza aperta**, in una piattaforma che rispetta i requisiti identificati in queste linee guida (3.4.1 *Scelta di uno strumento di code hosting*), registrandone poi il rilascio dentro Developers Italia (3.7 *Sviluppo di software ex novo*).

#### Fase di riuso

1. L'amministrazione «B», che necessita di un simile software, durante la fase di valutazione comparativa, trova il *software* messo a riuso dall'amministrazione «A», all'interno di Developers Italia (2.5.1 Fase 2.1: *Ricerca soluzioni riusabili per la PA*).
2. La licenza aperta consente all'amministrazione «B» di acquisire ed utilizzare il *software* dell'amministrazione «A» senza necessità di sottoscrivere alcuna convenzione, sottostando ai termini della licenza stessa.
3. L'amministrazione «B» effettua una valutazione dello stato del *software* e dell'applicabilità al proprio contesto (2.5.2 Fase 2.2: *Valutazione soluzioni riusabili per la PA*), inclusa l'eventuale necessità di una personalizzazione.
4. Se il *software* viene personalizzato, ove possibile, tale personalizzazione (in quanto sviluppo su specifica indicazione dell'amministrazione «B») è anch'essa soggetta a quanto prescritto dall'art. 69 comma 1, ed è quindi necessario rilasciare il relativo codice sorgente sotto **licenza aperta** (3.9 *Riuso di un software o utilizzo di un software Open Source*).

Il modello del riuso tramite *software* Open Source consente quindi di trovare un software, valutarlo e personalizzarlo **senza stipulare alcuna convenzione** con l'amministrazione che ha messo a riuso il software stesso, oltre all'accettazione della licenza Open Source che si perfeziona con il semplice download. Inoltre, il *software* è disponibile online e non è quindi necessaria alcuna richiesta di accesso.

È importante però considerare che il *software* potrebbe non essere «pronto all'uso». L'amministrazione potrebbe quindi avere necessità di un intervento tecnico per installare il software, adattarlo alle proprie esigenze, formare il personale che dovrà usarlo, avere a disposizione supporto e manutenzione. Per tutti questi interventi, l'amministrazione **può usare proprie risorse o forniture**, poiché nessun vincolo da questo punto di vista è imposto all'amministrazione che ha realizzato il software e lo ha messo a riuso.

### 3.3 Developers Italia e la ricerca di *software* in riuso

Il modello di riuso sopra delineato è reso possibile dalla piattaforma Developers Italia di AgID.

All'interno della piattaforma, viene resa disponibile una sezione dedicata al *software* reso disponibile per il riuso dalle amministrazioni. In particolare:

- È disponibile un «motore di ricerca» del *software* in riuso. Tramite questo motore, l'amministrazione potrà cercare *software* libero rilasciato su Internet da altre amministrazioni, utilizzando strumenti di consultazione messi a disposizione dalla piattaforma (es: filtri per tipologia di software, tipologia di amministrazione di riferimento);
- È disponibile una modalità per «registrare» in Developers Italia il *software* delle amministrazioni rilasciato in modalità Open Source ai fini del riuso, perché diventi facilmente individuabile da parte di altre amministrazioni.

### 3.4 Processo di messa a riuso del *software* sotto licenza aperta

Il processo di messa a riuso è il seguente:

1. L'amministrazione individua uno strumento di **strumento di code hosting**. Una volta identificato lo strumento, può essere utilizzato per tutto il *software* che deve essere messo a riuso (3.4.1 *Scelta di uno strumento di code hosting*).
2. L'amministrazione sceglie una licenza aperta da utilizzare (3.5 *Licenze aperte e scelta di una licenza*).
3. L'amministrazione, utilizzando proprie risorse oppure tramite un appalto, pubblica il codice sorgente completo del *software* e la relativa documentazione tecnica sullo strumento di code hosting. Questo processo tecnologico è descritto in *Allegato A: Guida alla pubblicazione di software come open source*, allegata a queste linee guida. La guida è scritta in modo da poter essere allegata ad un capitolato tecnico di gara, per facilitare l'acquisizione di un servizio demandando al fornitore gli adempimenti richiesti dalle presenti linee guida.

4. L'amministrazione «registra» il *software* sulla piattaforma Developers Italia, così che sia indicizzato dal motore di ricerca e reso visibile alle altre amministrazioni che cercano *software* in riuso.

Il processo qui delineato è valido sia per il *software* esistente di proprietà delle amministrazioni (3.6 *Rilascio di software esistente sotto licenza aperta*), sia per il *software* che verrà realizzato in futuro (3.7 *Sviluppo di software ex novo*).

### 3.4.1 Scelta di uno strumento di code hosting

Il rilascio di un *software* deve avvenire mediante uno strumento di code hosting, specializzato nell'ospitare e mettere a disposizione il software distribuito sotto licenza aperta. Esistono numerose soluzioni sul mercato, sia gratuite sia commerciali.

Poiché il fine del comma 1 dell'articolo 69 è quello di favorire il riuso tra amministrazioni, è necessario che lo strumento segua le best-practice in termini di funzionalità per la pubblicazione del codice sorgente, onde non causare costi aggiuntivi alle amministrazioni che vogliono trovare ed utilizzare il software.

In particolare, lo strumento dovrà necessariamente avere almeno le seguenti funzionalità:

- Accesso libero in lettura al codice sorgente, senza autenticazione;
- Registrazione gratuita e libera, aperta al pubblico;
- Interfaccia web per la lettura e navigazione del codice e della relativa documentazione;
- Utilizzo di un sistema di controllo di versione con la funzionalità di gestione di rami paralleli di sviluppo (*branch*);
- Sistema di segnalazioni (*issue tracker*) aperto al pubblico in lettura senza autenticazione e in scrittura dietro autenticazione;
- Implementazione di almeno un flusso di invio modifiche, revisione del codice (*code review*), e integrazione della modifica, completamente gestito dallo strumento, aperto al pubblico;
- Sistema di gestione dei rilasci;
- Disponibilità di API per interfacciarsi con lo strumento ed estrarre dati e metadati relativi ai repository.

Per semplificare la scelta, in *Allegato A: Guida alla pubblicazione di software come open source* è possibile trovare un elenco non esaustivo delle principali piattaforme sul mercato che corrispondono ai requisiti richiesti.

Alcune piattaforme completamente aderenti ai parametri minimi sono disponibili in modalità SaaS (cioè possono essere usate direttamente via Internet senza doverne installare una copia su un server), senza alcun costo di licenza, e senza la necessità di sottoscrivere contratti o convenzioni; la scelta di una di queste piattaforme SaaS è quindi da considerarsi preferenziale, nel caso non ci siano altri vincoli tecnici (es: requisiti di integrazione), in modo da non causare costi diretti o indiretti all'amministrazione.

L'amministrazione deve scegliere uno strumento (o più strumenti, in caso di motivate ragioni organizzative) sul quale effettuare i rilasci di tutto il *software* di cui è titolare. In alternativa, in *Allegato A: Guida alla pubblicazione di software come open source* è delineato un processo alternativo per demandare la scelta al soggetto incaricato di effettuare lo sviluppo del *software* e/o il rilascio dello stesso per conto dell'amministrazione (sia nel caso che questo soggetto corrisponda ad una risorsa interna o esterna all'amministrazione).

Una volta eletto uno o più strumenti di code hosting, l'amministrazione deve dare adeguata visibilità agli strumenti nella propria pagina istituzionale, come dettagliato sempre all'interno della medesima guida (*Registrazione del repository su Developers Italia*).

### 3.4.2 Registrazione del software aperto su Developers Italia

Il *software* rilasciato dalla amministrazione deve essere «registrato» all'interno del motore di ricerca di Developers Italia, per agevolare la consultazione alle altre amministrazioni che cercano un *software* in riuso.

Il processo tecnico preciso per effettuare la registrazione è indicato nella sezione *Registrazione del repository su Developers Italia*.

### 3.4.3 Responsabilità connesse al rilascio

L'amministrazione titolare del *software* non contrae alcun obbligo specifico legato al rilascio: non è infatti necessario fornire alcuna garanzia sul software, supporto tecnico o a livello utente, né tantomeno supportare economicamente le amministrazioni che riusano il *software* nei costi o nelle procedure di adozione.

## 3.5 Licenze aperte e scelta di una licenza

Per effettuare il rilascio del codice sorgente di un *software* sotto licenza aperta, l'amministrazione deve scegliere un testo di licenza appropriata.

### 3.5.1 Contesto

È necessario considerare che il legislatore, nel redigere l'articolo 69, ha chiaramente indicato come obiettivo quello di **favorire il riuso** del *software* stesso tra più amministrazioni. È dunque importante che la prima considerazione in ordine di importanza nella scelta della licenza sia quella di **valutare l'impatto che il testo della licenza ha sulla possibilità di riuso** da parte di altre amministrazioni.

Fin dagli anni '80, il mondo della ricerca informatica e dell'industria ha prodotto numerosi testi di licenza per il *software* Open Source, con l'obiettivo di creare un modello di condivisione mondiale del software. Con il progressivo aumento della complessità degli applicativi, si è reso sempre più importante, a scopo anche esclusivamente economico e di efficienza, lavorare integrando componenti già pronti, piuttosto che cominciare ogni volta a sviluppare codice da capo.

### 3.5.2 Licenze per il *software* aperto

Una licenza aperta, così come intesa nell'Art. 69 del CAD, è una licenza che garantisca all'utente di un *software* le seguenti libertà:

- Libertà di eseguire il *software* come si desidera, per qualsiasi scopo, senza ulteriori costi o restrizioni;
- Libertà di studiare come funziona il *software* e di modificarlo in modo da adattarlo alle proprie necessità;
- Libertà di ridistribuire copie del software;
- Libertà di modificare il *software* e distribuirne pubblicamente le versioni modificate.<sup>1</sup>

L'accesso al codice sorgente, o parimenti al formato necessario per riprodurre e modificare un'opera, è un prerequisito per rispettare tali libertà.

Open Source Initiative<sup>2</sup> (OSI) è un'organizzazione internazionale, riconosciuta a livello mondiale per il proprio processo di certificazione delle licenze *software* che rispettano detti requisiti. La lista aggiornata delle licenze certificate da OSI è disponibile al seguente indirizzo (indice alfabetico): <https://opensource.org/licenses/alphabetic>

L'adempimento dell'Art. 69 del CAD, relativamente alla scelta della licenza, deve essere effettuato **scegliendo una licenza tra quelle certificate da Open Source Initiative**. In alternativa, l'amministrazione che volesse provvedere in autonomia a redigere un testo di una licenza d'uso, può usare tale testo solo previa certificazione da Open Source Initiative, onde verificarne l'aderenza ai principi del *software* aperto. Il processo di invio di una licenza per approvazione è dettagliato all'indirizzo: <https://opensource.org/approval>.

Si noti che per identificare univocamente un testo di licenza, è possibile utilizzare la categorizzazione SPDX<sup>3</sup>, che associa ad ogni licenza (o combinazione) un codice e un'espressione univoci. Un elenco aggiornato dei codici e dei rispettivi testi di licenza è disponibile all'indirizzo: <https://spdx.org/licenses/>.

---

<sup>1</sup> Stallman, The Free *software* Definition - <https://www.gnu.org/philosophy/free-sw.it.html>

<sup>2</sup> <https://www.opensource.org/>

<sup>3</sup> <https://spdx.org>



In allegato alle linee guida (*Allegato C: Guida alle licenze Open Source*) si trova una guida per approfondire il tema delle licenze Open Source, che delinea una categorizzazione dei principali tipi di licenza e delle loro caratteristiche.

### 3.5.3 Scelta di una licenza

Una licenza di *software* libero consente l'utilizzo gratuito del codice sorgente cui si riferisce, dettando però alcuni vincoli da rispettare. Pertanto, l'integrazione di più componenti di *software* libero rilasciati sotto licenze diverse richiede una analisi di compatibilità delle stesse. Tale analisi può risultare eccessivamente complessa se le licenze coinvolte sono molteplici, comportando costi aggiuntivi.

In altre parole, **una proliferazione di licenze diverse rende più difficile e oneroso il riuso del software**, contravvenendo agli obiettivi delineati dall'art. 69 del CAD.

Si propone quindi il seguente albero decisionale per la scelta di una licenza aperta:

- Se il rilascio del *software* si riferisce ad una modifica di software Open Source esistente (quindi *software* preso a riuso da un'altra amministrazione o di proprietà di terze parti), l'amministrazione utilizzerà la **stessa licenza** con la quale è stato originariamente distribuito il software, per favorire la massima interoperabilità e riuso con altri utilizzatori del medesimo software;
- Se si tratta di un **software nuovo**, tranne per le eccezioni specificate sotto, utilizzare la licenza EUPL v1.2 (codice SPDX: EUPL-1.2): <https://spdx.org/licenses/EUPL-1.2.html>. Questa licenza, scritta dalla Commissione europea, è stata scelta in quanto di tipo «copyleft», garantisce massima interoperabilità a livello europeo, ed è anche tradotta in italiano. Sono previste solo alcune eccezioni a questa indicazione generale:
  - se il **software viene utilizzato principalmente via rete (es: tramite un browser)**, utilizzare la licenza «GNU Affero General Public License» versione 3 e successive (codice SPDX: AGPL-3.0-or-later): <https://spdx.org/licenses/AGPL-3.0-or-later.html>;  
Questa licenza è stata scelta perché, oltre ad essere compatibile con la maggior parte delle licenze Open Source, obbliga chi modifica il codice a rilasciare i miglioramenti anche in caso esso venga utilizzato come parte di un servizio SaaS.
  - se vengono rilasciati **componenti software** enucleati e con ampio campo applicativo (per esempio, le «**librerie software**» e gli «**SDK**»), utilizzare la licenza «BSD 3-Clause» (codice SPDX: BSD-3-Clause) <https://spdx.org/licenses/BSD-3-Clause.html>;  
Questa licenza è stata scelta per garantire un utilizzo da parte di tutti gli attori quanto più libero possibile, permettendo di realizzare applicativi basati su queste librerie, rilasciabili sotto qualunque licenza. Questo genere di componenti *software* è utilizzato normalmente per favorire l'interoperabilità con le Pubbliche Amministrazioni, e trovano beneficio nella nascita di ecosistemi che includono applicativi di terze parti, inclusi *software* proprietari.
- Per la **documentazione tecnica** del software, utilizzare la licenza Creative Commons CC-BY 4.0 (codice SPDX: CC-BY-4.0) <https://spdx.org/licenses/CC-BY-4.0.html>. Questa licenza è stata scelta in quanto permette un riutilizzo semplice della documentazione e degli esempi di codice in essa contenuta.

Si rimanda a *Allegato A: Guida alla pubblicazione di software come open source* per i dettagli tecnici su come apporre correttamente il testo di una licenza al codice sorgente nel momento della pubblicazione.

Le licenze scelte hanno un vasto utilizzo nell'ecosistema Open Source, dunque si massimizza la possibilità di poter integrare componenti di terze parti rilasciate con licenze compatibili.

L'amministrazione che volesse operare una scelta di licenza diversa da quella qui delineata deve motivarne le ragioni, analizzando la compatibilità tra le licenze adottate e quelle qui proposte, escludendo che la scelta limiti le opportunità di riuso ed assicurandosi che non comporti oneri aggiuntivi per le amministrazioni in fase di riuso.

## 3.6 Rilascio di *software* esistente sotto licenza aperta

Il comma 1 dell'Art. 69 recita:

»Le pubbliche amministrazioni che siano titolari di soluzioni e programmi informatici realizzati su specifiche indicazioni del committente pubblico, hanno l'obbligo di rendere disponibile il relativo codice sorgente, completo della documentazione e rilasciato in repertorio pubblico sotto licenza aperta, in uso gratuito ad altre pubbliche amministrazioni o ai soggetti giuridici che intendano adattarli alle proprie esigenze, salvo motivate ragioni di ordine e sicurezza pubblica, difesa nazionale e consultazioni elettorali.»

Gli obblighi qui citati si riferiscono all'intero parco *software* sul quale insistono i diritti di un'amministrazione con la conseguenza che, a prescindere dall'esigenza di rispettare tali obblighi in occasione del perfezionamento di nuovi contratti, ogni amministrazione è tenuta a darvi tempestiva attuazione **anche con riferimento al \*software\* già esistente ove sia titolare dei relativi diritti di proprietà intellettuale e industriale** (come indicato in *1.5 Titolarità*).

Dare attuazione a tali obblighi sul *software* già esistente costituisce un aspetto essenziale per la massimizzazione dell'efficacia della disposizione in commento e, più in generale, della buona prassi del riuso, giacché consente a altre amministrazioni di beneficiare senza ritardo delle opportunità offerte dal riuso, scongiurando il rischio che queste ultime si trovino a dover ri-acquistare soluzioni già appartenenti al patrimonio informativo pubblico e che, dunque, potrebbero essere utilizzate senza generare alcun ulteriore costo per la comunità.

È quindi **necessario** che l'amministrazione **provveda a censire il software** di cui è già in possesso al fine di verificarne la titolarità, e in caso positivo proceda al rilascio sotto licenza aperta.

Vista la rapida obsolescenza delle tecnologie digitali, e considerata l'importanza di favorire il riuso delle soluzioni disponibili, si ritiene escluso dall'obbligo di rilascio il *software* che non sia più in uso da parte dell'amministrazione da oltre 5 anni, con riferimento alla data di pubblicazione delle presenti linee guida.

Le specifiche di dettaglio su come effettuare il rilascio sono contenute in *Allegato A: Guida alla pubblicazione di software come open source*. Laddove l'amministrazione non avesse le risorse necessarie per allineare la documentazione agli standard previsti della Guida, l'amministrazione deve procedere comunque immediatamente al rilascio di quanto abbia in possesso nello stato in cui si trova, fermo restando che la presenza di documentazione è requisito essenziale previsto dalla norma e sarà quindi necessario procedere al completamento e allineamento della documentazione al più presto per considerare l'adempimento concluso.

### 3.7 Sviluppo di *software* ex novo

Se l'amministrazione, seguendo la valutazione comparativa proposta in *2 Linee Guida sull'acquisizione di software*, sceglie di sviluppare un *software* ex novo ricorrendo a risorse interne oppure tramite appalto, è importante adempiere a quanto previsto nell'articolo 69, e in particolare:

- al comma 1, che richiede il rilascio sotto licenza aperta per consentire il riuso da parte di altre amministrazioni;
- al comma 2, che richiede che l'amministrazione acquisisca la titolarità del *software* che viene realizzato;
- al comma 2-bis, che delega alle presenti Linee guida la scelta delle piattaforme di pubblicazione del codice sorgente.

Le seguenti sezioni approfondiscono come adempiere alle disposizioni richiamate.

#### 3.7.1 Rilascio di nuovo *software* sotto licenza aperta

È importante che il comma 1 dell'articolo 69, che richiede il rilascio sotto licenza aperta, venga considerato fin dall'inizio dello sviluppo e non solo al termine dell'intero processo. I requisiti tecnici per il rilascio sono descritti in *Allegato A: Guida alla pubblicazione di software come open source*.

I costi sostenuti per effettuare quanto descritto nella guida saranno notevolmente inferiori se le specifiche tecniche descritte saranno seguite fin dall'inizio dello sviluppo.

In caso di appalto, **si richiede che le amministrazioni inseriscano sempre** la *Allegato A: Guida alla pubblicazione di software come open source* **tra i documenti di gara**, per esempio in allegato al capitolato tecnico.

Viceversa, l'amministrazione che procedesse tardivamente all'adempimento del comma 1 art. 69, utilizzando una gara successiva al completamento della prima, incorrerebbe in maggiori oneri economici.

Pertanto, si suggerisce di sviluppare il *software* direttamente sullo strumento di code hosting selezionato, fin dalle prime fasi di progettazione, senza attendere la versione preliminare per effettuare il rilascio.

### 3.7.2 Acquisizione della titolarità di *software* sviluppato ex novo

Come già discusso in *1.5 Titolarità*, l'amministrazione deve assicurarsi la piena titolarità del *software* realizzato *ex novo*. Si rimanda al citato paragrafo per ulteriori informazioni.

## 3.8 Manutenzione di un *software* da parte dell'amministrazione titolare

La manutenzione di un *software*, sia essa evolutiva o correttiva, è un processo essenziale del ciclo di vita, poiché mantiene il *software* aggiornato relativamente alla veloce evoluzione tecnologica, all'evoluzione normativa, e alle nuove esigenze dell'amministrazione.

Durante la manutenzione, inoltre, gli aggiornamenti del *software* prodotti entrano nel campo di applicazione dell'art. 69 del CAD e devono quindi essere messi a riuso. La presente sezione descrive la procedura manutentiva che indichiamo per consentire in modo agevole il riuso di questi miglioramenti.

### 3.8.1 Titolarità del codice sviluppato in fase di manutenzione

Come già discusso in *1.5 Titolarità*, l'amministrazione deve assicurarsi la piena titolarità del *software* realizzato *ex novo*, ivi compresa ogni porzione realizzata durante un contratto di manutenzione. Si rimanda al citato paragrafo per ulteriori informazioni.

### 3.8.2 Rilascio sotto licenza aperta delle modifiche

Per rilasciare le modifiche ad un *software*, non è possibile utilizzare il processo descritto precedentemente in *3.7.1 Rilascio di nuovo software sotto licenza aperta*; tale processo infatti, indipendentemente dall'entità della modifica, creerebbe un secondo repository di codice sorgente disgiunto dall'originale, causando costi elevati per qualunque amministrazione che, avendo preso in riuso il *software* originale, voglia continuare a beneficiare della sua evoluzione.

Il modo corretto e con minori oneri (sia per l'amministrazione titolare, sia per quelle che vorranno riutilizzare il *software* in futuro) per mantenere un *software* sotto licenza aperta è quello di adottare uno **specifico processo di sviluppo** nel quale ogni singola modifica venga effettuata direttamente nel *repository* originale contenente il *software*, dando immediata evidenza del cambiamento avvenuto.

Inoltre, è necessario comunicare che il *software* è in fase di manutenzione (inserendo tale informazione anche all'interno della registrazione del *software* in Developers Italia), affinché altre amministrazioni possano tenerne conto nella fase di valutazione comparativa.

Il processo completo è descritto, nei suoi dettagli tecnici, in *Allegato B: Guida alla manutenzione di software open source*. In caso di appalto, si richiede che l'amministrazione **alleghi la Guida tra i documenti tecnici di gara**, per esempio come allegato al capitolato tecnico, in modo che i fornitori abbiano evidenza immediata del processo richiesto.

### 3.8.3 Supporto alle amministrazioni che riusano

Anche se non sussiste nessun obbligo di garanzia o supporto tecnico o formativo da parte dell'amministrazione titolare verso le amministrazioni che prendono in riuso, ove il *software* sia soggetto a manutenzione evolutiva, si richiede che le risorse interne o le aziende incaricate di tale manutenzione offrano un supporto base a chi segnala in modo circostanziato eventuali anomalie, oppure voglia effettuare (a proprie spese) modifiche al software.

Il modello di riuso, infatti, consente a più amministrazioni di investire sul medesimo software, ciascuna con il proprio *budget*, andando quindi a costruire un valore incrementale sul *software* originario. Affinché detto processo funzioni correttamente, però, è auspicabile almeno un coordinamento tecnico tra la Pubblica Amministrazione che mantiene il *software* e quella che intenda modificarlo. Inoltre, questo offre una opportunità di condividere piani di sviluppo e dunque investimenti tra più amministrazioni sul medesimo software, con risparmio per la finanza pubblica.

Anche questo processo di supporto alla modifica di un *software* è dettagliato nella stessa guida *Allegato B: Guida alla manutenzione di software open source*.

### 3.8.4 Software non ancora rilasciato sotto licenza aperta

Se l'amministrazione avvia un processo di manutenzione di un software che già possiede, ma per il quale non abbia ancora provveduto al rilascio sotto licenza aperta, si deve valutare l'aggiunta dell'attività di primo rilascio al contratto di manutenzione, in ragione del minor costo che normalmente si sostiene rispetto ad effettuarlo separatamente.

## 3.9 Riutilizzo di un software o utilizzo di un software Open Source

Nel secondo capitolo delle presenti Linee guida, dedicato all'art. 68, viene illustrato il processo con il quale l'amministrazione decide la modalità di individuazione del software.

Se l'amministrazione, seguendo la valutazione comparativa proposta, sceglie di prendere a riuso un *software* esistente o di utilizzare un software Open Source, il processo di utilizzo viene descritto in questa sezione, ed è il medesimo nei due casi.

### 3.9.1 Utilizzo di software a riuso o Open Source

In generale, **non è necessario conseguire un'autorizzazione da parte del titolare dei diritti sul software**; il modello di riuso delineato dall'uso delle licenze aperte, infatti, consente di **adottare un software senza necessità di stipulare alcuna convenzione, né una richiesta di accesso**: il *software* pubblicato secondo quanto descritto in *3.6 Rilascio di software esistente sotto licenza aperta* è immediatamente fruibile per una analisi della compatibilità alle esigenze, per la personalizzazione e per l'utilizzo.

Come è spiegato in *3.8.3 Supporto alle amministrazioni che riusano*, è consigliabile entrare in contatto con l'attuale incaricato alla manutenzione del software, per concordare tecnicamente come effettuare nel modo più efficace le modifiche necessarie e coordinare gli sforzi economici.

### 3.9.2 Modifiche ad un software a riuso o Open Source

Dal punto di vista normativo, le modifiche o personalizzazioni ad un software sotto licenza aperta sono soggette all'art. 69 comma 2 e devono essere quindi effettuate acquistando piena titolarità del codice sviluppato. Il riuso di *software* senza apporto di modifiche, invece, non configura l'obbligo di rilascio.

Sotto il profilo di acquisizione della titolarità, il fatto che il software oggetto di modifica non sia di proprietà dell'amministrazione che effettua la modifica non esime quest'ultima dall'obbligo di acquisire la titolarità delle modifiche sviluppate. Si rimanda quindi a *1.5 Titolarità*.

Viceversa, a livello tecnico, il processo per effettuare le modifiche è diverso dal processo di manutenzione descritto in *3.8 Manutenzione di un software da parte dell'amministrazione titolare* poiché gli interventi avverranno su un

*software* del quale non si ha la piena titolarità e dunque è auspicabile un coordinamento tecnico, già descritto a livello di opportunità e benefici in *3.8.3 Supporto alle amministrazioni che riusano*.

Il processo tecnico è dettagliato in *Allegato D: Guida alla presa in riuso di software open source*. In caso di appalto, si richiede che l'amministrazione **alleghi la Guida tra i documenti tecnici di gara**, per esempio come allegato al capitolato tecnico, in modo che i fornitori abbiano evidenza immediata del processo richiesto.



---

## Allegato A: Guida alla pubblicazione di *software* come open source

---

Questa guida è rivolta alle Amministrazioni che, titolari di un software, lo vogliono rilasciare in modalità open-source (codice a sorgente aperta). La guida può essere utilizzata da chiunque sia incaricato di svolgere le attività in essa descritte: le risorse interne dell'Amministrazione, società in-house dell'Amministrazione, un fornitore di servizi individuato dall'Amministrazione. Nel corso della descrizione delle attività si utilizza il termine “**Incaricato**” indistintamente per tutte e tre queste categorie.

La guida è stata prodotta anche per poter essere allegata ad un capitolato tecnico nell'ambito di un appalto; in questo caso l'Incaricato è tenuto a svolgere le attività descritte nel presente documento come parte integrante dell'appalto, in aggiunta a quanto specificato nel resto del capitolato.

Nel documento sarà adottata la seguente convenzione:

- **MUST/MUST NOT**: prescrizioni obbligatorie che l'Incaricato è tenuto a rispettare;
- **SHOULD/SHOULD NOT**: raccomandazioni che l'Incaricato è tenuto a valutare ed attuare qualora non vi siano documentabili ragioni ostative;
- **MAY/MAY NOT**: scelte che l'Incaricato può attuare a propria discrezione.

### A.1 Premessa

Il presente documento illustra le modalità tecniche con le quali un software di proprietà di una Pubblica Amministrazione è rilasciato in modalità open source (codice a sorgente aperto). Le attività di seguito riportate sono attribuite al soggetto Incaricato dalla Pubblica Amministrazione titolare del codice.

Il contesto normativo è il seguente:

- L'art. 69 comma 1 del Codice dell'Amministrazione Digitale impone che  
«Le pubbliche amministrazioni che siano titolari di soluzioni e programmi informatici realizzati su specifiche indicazioni del committente pubblico, hanno l'obbligo di rendere disponibile il relativo codice sorgente, completo della documentazione e rilasciato in repertorio pubblico sotto licenza aperta, in uso gratuito ad altre pubbliche amministrazioni o ai soggetti giuridici che intendano adattarli alle proprie esigenze, salvo motivate ragioni di ordine e sicurezza pubblica, difesa nazionale e consultazioni elettorali.»
- Le Linee Guida di AgID (di seguito «Linee Guida») forniscono approfondimenti su questo obbligo, chiarendo il modello di riuso delineato dalla legge e definendo i parametri principali per la scelta della licenza e il rilascio del codice.

Le modalità descritte nel presente documento si ispirano alle best practice adottate nello sviluppo open source. Oltre alle prescrizioni qui espresse si consiglia la consultazione della guida <https://opensource.guide> per suggerimenti su come impostare correttamente il lavoro.

## A.2 Individuazione dello strumento di code hosting

L'Amministrazione titolare del *software* deve individuare uno strumento di code hosting da utilizzare per il rilascio.

Le Linee Guida di AgID specificano i parametri tecnici minimi, qui riportati nuovamente:

- Accesso libero in lettura al codice sorgente, senza autenticazione;
- Registrazione gratuita e libera, aperta al pubblico;
- Interfaccia web per la lettura e navigazione del codice e della relativa documentazione;
- Utilizzo di un sistema di controllo di versione con la funzionalità di gestione di rami paralleli di sviluppo (*branch*);
- Sistema di segnalazioni (*issue tracker*) aperto al pubblico in lettura senza autenticazione e in scrittura dietro autenticazione;
- Implementazione di almeno un flusso di invio modifiche, revisione del codice (*code review*), e integrazione della modifica, completamente gestito dallo strumento, aperto al pubblico;
- Sistema di gestione dei rilasci;
- Disponibilità di API per interfacciarsi con lo strumento ed estrarre dati e metadati relativi ai repository.

Sono conformi a questi requisiti, e raccomandati in virtù della loro maturità e diffusione internazionale, i seguenti strumenti di code hosting:

- GitHub - <https://github.com/> (gratuita);
- BitBucket - <https://bitbucket.org/> (gratuita o self-hosted a pagamento);
- GitLab - <https://gitlab.com/>
- Phabricator/Phacility - <https://www.phacility.com/>
- Gitea - <https://gitea.io/>
- Gogs - <https://gogs.io/>

Tale elenco è da considerare esemplificativo e non esaustivo.

L'Amministrazione titolare dei diritti indicherà all'Incaricato uno strumento di code hosting da utilizzare per il rilascio; in mancanza di tale indicazione, l'Incaricato può proporre uno strumento di sua preferenza.

È richiesto che la piattaforma individuata sia disponibile e mantenuta indipendentemente dalla corrente lavorazione, ovvero sia erogata in modalità SaaS da terze parti o sia istituita dall'Amministrazione committente o da un'altra Amministrazione per finalità più generali rispetto al presente progetto (MUST).

Qualora il *software* costituisca un lavoro derivato di altro software open source già esistente, è preferibile adottare la medesima piattaforma in modo da sfruttarne le funzionalità di collaborazione (SHOULD).

Se l'Amministrazione dispone già di una propria area («organizzazione») presso lo strumento di code hosting individuato, sarà concesso l'accesso ai referenti dell'Incaricato. In caso contrario, sarà l'Incaricato ad aprire un account presso lo strumento concordato; il nome dell'area dovrà rispecchiare il progetto e non la denominazione dell'Amministrazione, né fare riferimento all'Incaricato; inoltre, l'Incaricato dovrà fornire all'Amministrazione l'accesso allo strumento con i massimi poteri. L'Amministrazione resterà proprietaria dell'area anche dopo il termine della presente lavorazione.

All'interno dello strumento scelto, l'Incaricato aprirà un repository destinato ad ospitare il *software* in sviluppo. Qualora la lavorazione sia articolata in più componenti logicamente distinti e con finalità autonome, purché singolarmente compiuti, documentati e riutilizzabili separatamente, dovranno essere aperti repository distinti (MUST).



Il link al repository dovrà essere riportato nell'interfaccia del software esposta al pubblico (ad es. con un link nel *footer* - piè di pagina - o tra le pagine di aiuto) in modo che sia possibile per l'utente reperire la versione del codice così come è in esecuzione.

## A.3 Scelta della licenza

La licenza aperta da adottare deve essere indicata dall'Amministrazione in sede di capitolato o concordata in esecuzione, conformemente alle Linee Guida. L'Incaricato è tenuto (MUST) a garantire la compatibilità di tale licenza con quelle di eventuali componenti riutilizzati od incorporati, con o senza modifiche, per i quali non si detengono i diritti (ad esempio: librerie, asset grafici), ivi inclusi quelli di proprietà dell'Incaricato stesso. Se tali componenti si trovano in file distinti, è possibile mantenere la licenza distinta purché ciò sia permesso dalle licenze e i relativi file riportino chiaramente l'indicazione della differente licenza e dei titolari dei diritti economici di sfruttamento (MAY).

## A.4 Attribuzione della licenza ed individuazione della titolarità

Al fine di applicare la licenza scelta al materiale da rilasciare è necessario creare nella root del repository un file denominato LICENSE, contenente il testo integrale della licenza scelta, senza alcuna modifica. I testi originali sono disponibili all'indirizzo <https://spdx.org/licenses/>. È obbligatorio (MUST) specificare la licenza applicata tramite espressione (o codice) SPDX all'inizio di ogni file sorgente, in modo che sia facilmente possibile una metadattazione automatica delle licenze usate.

Si consiglia la lettura della guida <https://reuse.software/practices/2.0/> per ulteriori raccomandazioni sull'applicazione della licenza a diversi formati di file.

Ai sensi dell'art. 69 comma 2 del Codice dell'Amministrazione Digitale il detentore dei diritti da indicare nel codice sorgente è l'Amministrazione committente (MUST), che ha acquisito la titolarità.

## A.5 Individuazione dei materiali da rilasciare

Sono soggetti all'obbligo di rilascio in open source i seguenti materiali:

- codice sorgente;
- struttura di database;
- script o altri materiali necessari all'installazione in ambiente di sviluppo o di produzione;
- asset grafici generici (ad es. bottoni, elementi grafici);
- documentazione finalizzata all'installazione delle dipendenze, alla compilazione (ove applicabile), alla messa in funzione.

Sono esclusi dall'obbligo di rilascio i seguenti materiali:

- dati utilizzati in produzione o trattati con il *software* sviluppato;
- asset grafici specifici (ad es. loghi di aziende) sui quali non sia applicabile la licenza scelta.

## A.6 Rilascio del codice e organizzazione del repository

Il codice sorgente deve essere rilasciato in versione integrale e senza omissioni in modo che un soggetto terzo possa, seguendo la documentazione, compilarlo (ove applicabile) e metterlo in funzione senza doverlo modificare. I nomi delle variabili, delle funzioni, delle classi e degli altri simboli devono essere mantenuti in chiaro e

devono essere comprensibili; parimenti, il codice non deve essere sottoposto ad alcun trattamento di compressione (c.d. *minification*) che ne ostacoli la leggibilità. Qualsiasi tentativo di offuscamento è considerato violazione dell'obbligo di rilascio.

Deve essere posta massima attenzione sulla leggibilità del codice, che deve essere correttamente indentato e commentato in ogni suo passaggio (MUST). È richiesta l'adozione di un *coding style* coerente e pulito. Alcuni esempi di convenzioni:

- <https://github.com/google/styleguide>
- <https://www.gnu.org/prep/standards/>
- <https://www.kernel.org/doc/Documentation/process/coding-style.rst>
- <http://www.php-fig.org/psr/psr-2/>
- <http://pear.php.net/manual/en/standards.php>

È raccomandata (SHOULD) l'adozione di un'architettura modulare, basata sulla suddivisione della logica in librerie specializzate e riutilizzabili singolarmente, con API interne definite e documentate nei commenti del codice. In caso di integrazione di librerie esterne, si raccomanda (SHOULD) l'uso dei *package manager*, per facilitare la manutenzione e l'aggiornamento.

Il rilascio in open source non deve essere considerato come mero adempimento da svolgersi al termine della lavorazione, ma deve essere previsto sin dalla fase di sviluppo ad esempio strutturando il software in modo che tutte le specificità dell'Amministrazione committente (nomi, indirizzi, server) siano modificabili attraverso file di configurazione (SHOULD) e che il *software* sia pronto al riuso da parte di altro soggetto.

Il repository deve essere organizzato con una struttura di directory chiara e comprensibile (MUST), ad esempio separando in directory distinte documentazione, librerie, eseguibili, script di servizio, test suite, eccetera.

## A.7 File README

Il repository deve contenere un file denominato README.md contenente:

- (MUST) il titolo del repository ed un sottotitolo descrittivo;
- (MUST) descrizione estesa del repository in un linguaggio comprensibile anche dai non addetti ai lavori (evitare acronimi e gergo tecnico), in particolare:
  - contesto di utilizzo e casi d'uso;
  - finalità del software;
  - screenshot (se il *software* dispone di interfaccia grafica, anche web);
  - link ad eventuali pagine istituzionali relative al progetto o al contesto di utilizzo;
- (MUST) link ad eventuale documentazione aggiuntiva non inclusa nel presente repository;
- (MUST) spiegazione struttura del repository anche a beneficio dei potenziali contributori (struttura delle directory e dei branch);
- (MUST) elenco dettagliato prerequisiti e dipendenze (sistemi operativi, librerie, framework eccetera) con esplicita indicazione di eventuali dipendenze da *software* commerciali;
- (MUST) istruzioni per l'installazione:
  - procedura di installazione di requisiti e dipendenze;
  - build system (se previsto dal progetto);
  - comandi per la compilazione o il deployment, possibilmente automatizzati da uno script/Makefile (se previsto dal progetto);
- (MUST) eventuali indicazioni sullo status del progetto:
  - stato di alpha/beta/stabile eccetera;

- importanti limitazioni o known issues;
- (SHOULD) link ad eventuali sistemi di Continuous Integration (TravisCI, CircleCI), code coverage (copertura del codice) ed altre metriche associati al repository;
- (SHOULD) documentazione relativa all'eventuale utilizzo di sistemi per semplificare e accelerare il *deployment* in ambiente di sviluppo, test e produzione (ad esempio immagini Docker o altri sistemi di virtualizzazione con predisposizione di immagini preconfigurate);
- (MUST) nomi dei detentori di copyright, ovvero l'Amministrazione committente;
- (MUST) nomi dei soggetti incaricati del mantenimento del progetto open source (è richiesto il nome dell'azienda e facoltativamente si possono aggiungere nomi delle persone incaricate);
- (MUST) indirizzo e-mail a cui inviare segnalazioni di sicurezza (specificare che le segnalazioni di sicurezza non vanno inviate attraverso l'issue tracker pubblico ma devono essere inviate confidenzialmente a tale indirizzo e-mail);

## A.8 Documentazione

È necessario (MUST) allegare al *software* la documentazione necessaria ad:

- installare le dipendenze;
- installare un ambiente di sviluppo da zero (meglio se corredata da script, immagini di *container*, Makefile o altri strumenti per rendere l'operazione rapida);
- compilare il *software* (ove applicabile);
- installare il *software* in ambiente di produzione;
- comprendere l'architettura del *software* (a beneficio di soggetti terzi che intendano riusarlo od integrarlo).

La documentazione allegata deve (MUST) inoltre seguire le indicazioni sul rilascio di documentazione tecnica prescritte nelle Linee Guida di design per i servizi web della Pubblica Amministrazione (sezione Content Design) e la Guida a Docs Italia, entrambe pubblicate da AgID. La documentazione deve essere scritta in un formato testuale che garantisca il versionamento riga per riga (ad esempio sono ammessi i seguenti formati: HTML, Markdown, reStructuredText, LaTeX). La documentazione in formato ODT, DOCX o PDF non è ammessa poiché si tratta di formati con i quali non è possibile definire le diverse versioni «riga per riga».

Se nel capitolato è prevista anche la stesura di documentazione sull'utilizzo del *software* rivolta agli utenti finali («manuale utente» o simile documento), l'obbligo di rilascio si estende anche ad essa. Per tale documentazione sono consentiti anche formati binari, purché aperti, modificabili e multiplatforma (resta dunque escluso il formato PDF).

## A.9 Tempi di rilascio

All'inizio della lavorazione, l'Incaricato concorda con l'Amministrazione il piano di rilascio in open source del *software* durante lo sviluppo. Le Linee Guida suggeriscono di adottare un modello di sviluppo aperto, che preveda il rilascio contestuale allo sviluppo sin dall'inizio. Questo modello consente anche ad altre amministrazioni di venire a conoscenza delle attività di sviluppo, anche prima della prima messa in produzione, diminuendo la probabilità che due amministrazioni sviluppino in modo indipendente *software* analoghi.

Qualora non si opti per un modello di sviluppo aperto, il rilascio in open source deve essere effettuato (MUST) entro 15 giorni dal momento dell'acquisizione del *software* da parte dell'Amministrazione committente al termine della lavorazione, ovvero dal momento in cui detto software viene immesso in collaudo o in produzione, ovvero da una richiesta dell'Amministrazione che può comunque essere trasmessa all'Incaricato in qualsiasi fase. Se la lavorazione è articolata in più lotti, i presenti termini di rilascio si applicano a ciascun lotto.

A partire dal momento del rilascio, qualsiasi successiva modifica deve essere pubblicata tempestivamente nel repository, indipendentemente dalla messa in collaudo o in produzione (MUST). Al fine di gestire tali flussi di

rilascio e collaudo l'Incaricato può usare le funzionalità di *branching* offerte dal sistema di controllo di versione prescelto (MAY).

### A.10 Sicurezza

Ricordando che la sicurezza del *software* è un tema importante di cui tenere conto durante il ciclo di sviluppo e che non verrà trattato in questo documento, si indicano qui alcuni principi base su attenzioni specifiche da adottare durante il processo di rilascio.

È necessario (MUST) rimuovere dal codice sorgente qualsiasi password o certificato o altra credenziale relativi a sistemi reali (anche di test); a tale scopo si deve ricorrere a file di configurazione separati o a blacklist nel sistema di controllo di versione (ad esempio, il file `.gitignore` o `.hgignore`). Qualora si intenda integrare il repository con un meccanismo di deployment automatico e dunque si necessiti di mantenere delle credenziali, è possibile utilizzare i meccanismi sicuri di cifratura previsti per la piattaforma di code hosting e per i sistemi di Continuous Integration adottati (ad es. `git-crypt`).

È importante verificare che non si siano depositate per errore tali credenziali (**API keys, secrets, password, ...**) all'interno del repository, non solo nella versione corrente ma anche in revisioni precedenti.

Deve essere evitata se possibile (MAY) la riscrittura di algoritmi già disponibili in librerie open source esterne (ad esempio: crittografia, sanitizzazione dell'input, protocolli di rete, parsing di XML o altri formati, gestione della memoria eccetera).

Tutto il codice «morto», ovvero non utilizzato, deve essere rimosso (MUST) poiché potrebbe portare a confusione od essere considerato mantenuto ed erroneamente reintegrato senza i necessari controlli.

Se il *software* è un'applicazione web esposta su rete pubblica, o contiene applicazioni web, dovrebbe (SHOULD) essere accessibile per ogni installazione al path `https://<hostname>/.well-known/security.txt` un file formattato secondo le indicazioni del sito <https://securitytxt.org>. Tale file è finalizzato a fornire informazioni utili a chi rilevi vulnerabilità ed intenda inviare segnalazioni di sicurezza.

### A.11 Registrazione del repository su Developers Italia

Non appena il repository pubblico è stato aperto, è necessario (MUST) effettuare la registrazione su Developers Italia, per garantire che venga indicizzato e presentato nel motore di ricerca presente sul sito.

La registrazione avviene seguendo due passaggi:

1. **Pubblicazione di un file `publiccode.yml` nella directory root del repository.** «`publiccode.yml`» è uno standard che identifica il progetto come «software utile per la Pubblica Amministrazione», e contemporaneamente offre una serie di informazioni utili alla valutazione del *software* stesso per il riuso. Tale file verrà rilevato automaticamente dall'indicizzatore (crawler) di Developers Italia al fine della generazione della relativa scheda nel catalogo. La documentazione sul formato può essere trovata qui: <https://github.com/italia/publiccode.yml>
2. **Aggiunta dello strumento di code-hosting al motore di ricerca.** Al fine di accertarsi che Developers Italia identifichi correttamente il repository come di proprietà della pubblica amministrazione, è necessario registrare lo strumento di code-hosting (o meglio, la «organizzazione» all'interno dello stesso) la prima volta che viene usato, associandolo alla Pubblica Amministrazione. La procedura è da seguire è dettagliata qui: <https://onboarding.developers.italia.it>

---

### Allegato B: Guida alla manutenzione di *software* open source

---

Questa guida è rivolta alle Amministrazioni che, titolari di un *software* già pubblicato in open source, vogliono effettuare una manutenzione sullo stesso. La guida può essere utilizzata da chiunque sia incaricato di svolgere le attività in essa descritte: le risorse interne dell'Amministrazione, società in-house dell'Amministrazione, un fornitore di servizi individuato dall'Amministrazione. Nel corso della descrizione delle attività si utilizza il termine "Incaricato" indistintamente per tutte e tre queste categorie.

La guida è stata prodotta anche per poter essere allegata ad un capitolato tecnico nell'ambito di un appalto; in questo caso l'Incaricato è tenuto a svolgere le attività descritte nel presente documento come parte integrante dell'appalto, in aggiunta a quanto specificato nel resto del capitolato.

Nel documento sarà adottata la seguente convenzione:

- **MUST/MUST NOT:** prescrizioni obbligatorie che l'Incaricato è tenuto a rispettare;
- **SHOULD/SHOULD NOT:** raccomandazioni che l'Incaricato è tenuto a valutare ed attuare qualora non vi siano documentabili ragioni ostative;
- **MAY/MAY NOT:** scelte che l'Incaricato può attuare a propria discrezione.

#### B.1 Obbligo di rilascio

Quando, nell'ambito delle attività di manutenzione, si apportano modifiche al codice originale, anche di carattere minore, si configura l'obbligo di rilascio ex art. 69 del Codice dell'Amministrazione Digitale.

Qualora l'Amministrazione sia già titolare di un repository destinato al software oggetto della manutenzione, creato secondo le indicazioni della *Guida alla pubblicazione di \*software open source\**, il rilascio delle modifiche andrà effettuato mediante aggiornamento di tale repository prima che le stesse vengano immesse in collaudo o in produzione (MUST). Al fine di gestire tali flussi di rilascio e collaudo, distinguendo la versione già in produzione da quella in sviluppo o in collaudo, l'Incaricato può usare le funzionalità di *branching* offerte dal sistema di controllo di versione prescelto (MAY).

Qualora invece l'Amministrazione non sia già titolare di un repository per il *software* oggetto della manutenzione, dovrà procedere a crearne uno seguendo le indicazioni della *Guida alla modifica di \*software open source di terzi\**.

## B.2 Obblighi relativi alla manutenzione di *software* per il quale l'Amministrazione disponga già di un repository

Le disposizioni successive si applicano ove l'Amministrazione sia titolare di un repository.

### B.2.1 Aggiornamento delle dipendenze

Per tutta la durata dell'incarico di manutenzione, l'Incaricato è tenuto a monitorare i rilasci delle eventuali dipendenze incorporate nel software e a recepire eventuali aggiornamenti (MUST). Se il *software* è derivato da altro software, tale obbligo di monitoraggio e recepimento si applica anche al *software* originale (c.d. *upstream*).

Eventuali incompatibilità o problemi di sicurezza insorti nel tempo dovranno essere documentati tempestivamente attraverso l'apertura di issue dedicate, da tenere aperte fino alla risoluzione, ed eventualmente anche nel file README. Nel caso di nuove versioni che risolvono problemi di sicurezza, l'aggiornamento delle dipendenze deve avere priorità assoluta.

### B.2.2 Descrizione del ruolo di maintainer

All'interno di un progetto open source, il maintainer è il soggetto che svolge un'attività di controllo e direzione degli sviluppi sul progetto, e a cui la *community* che afferisce al *software* (es: utilizzatori) può segnalare problematiche o discutere miglioramenti.

Per tutta la durata dell'attività di manutenzione connessa al software, l'Amministrazione titolare svolgerà il ruolo di *maintainer* del progetto open source, affidandone l'esecuzione all'Incaricato, il quale inserirà il nome della propria azienda o ente e i riferimenti di contatto nei file README e *publiccode.yml* del repository, con l'eventuale data di termine dell'incarico. L'Incaricato dovrà quindi, per conto dell'Amministrazione, gestire l'attività sul progetto derivante dalle interazioni con gli utenti esterni.

### B.2.3 Interazione sul repository/issue tracker

Tutte le interazioni avviate da utenti esterni all'interno della piattaforma di code hosting, e in particolare attraverso il suo issue tracker, dovranno essere esaminate dall'Incaricato entro due giorni lavorativi (SHOULD), ed entro tale termine è necessario (MUST) fornire una risposta. La risposta può non essere esaustiva, e laddove non sia possibile rispondere approfonditamente subito è comunque opportuno dare un cortese riscontro con delle prime considerazioni.

### Risoluzione di bug

Le segnalazioni di bug ricevute da utenti esterni attraverso il sistema di *issue tracking* dovranno essere analizzate al pari di quelle ricevute dall'Amministrazione committente. Se la risoluzione fosse compatibile (in quanto a tempi e costi) con le attività previste dal contratto, potrà essere eseguita senza necessità di ulteriore approvazione. Se invece la risoluzione non fosse compatibile (in quanto a tempi e costi) con le attività di manutenzione prevista dal contratto, la issue dovrà essere mantenuta aperta, informando l'Amministrazione titolare della propria scelta.

Il processo di diagnosi e risoluzione dovrà essere documentato pubblicamente all'interno dell'issue tracker, ad eccezione delle informazioni che hanno implicazioni sulla sicurezza dei sistemi in produzione, le quali dovranno essere mantenute riservate fino alla messa in funzione delle correzioni (MUST) e solo successivamente pubblicate (MUST), a beneficio di altri utenti del software. La issue di segnalazione dovrà essere mantenuta aperta fino alla risoluzione (MUST) ed è opportuno (SHOULD) chiedere all'utente originale di verificare in prima persona la bontà della risoluzione prima di chiuderla. In caso di mancata risposta dell'utente per trenta giorni, l'Incaricato può chiudere la issue, dopo aver documentato all'interno l'avvenuto collaudo della modifica.

### Richieste di nuove funzionalità

Le richieste di nuove funzionalità dovranno essere valutate dall'Incaricato, di concerto con l'Amministrazione, in relazione alla loro pertinenza al progetto. Se non ritenute pertinenti dovranno essere chiuse (SHOULD) fornendo una motivazione al proponente.

Se ritenute pertinenti dovranno essere lasciate aperte fino all'eventuale implementazione (MUST), dando tuttavia rapido riscontro al proponente (MUST) con una valutazione sulla fattibilità tecnica della richiesta e suggerimento su eventuali altri modi per raggiungere l'obiettivo dichiarato. L'Incaricato può chiedere al proponente, se necessario, maggiori dettagli sul caso d'uso che motiva la richiesta (MAY).

L'implementazione delle funzionalità richieste dovrà essere approvata dall'Amministrazione (MUST) nel caso che questo comporti degli oneri per la stessa (es: in caso il contratto sia strutturato con un modello a consumo).

In alternativa, l'Incaricato può decidere in autonomia di dare seguito alla richiesta implementandola nel codice (MAY), senza causare oneri aggiuntivi all'Amministrazione e nel rispetto dei tempi del contratto (per esempio, in virtù di altri accordi commerciali sullo stesso software).

### Richieste di informazioni o supporto

Le richieste di informazioni sul progetto dovranno essere evase a cura dell'Incaricato entro 2 giorni lavorativi (SHOULD). Le risposte dovranno limitarsi alle caratteristiche tecniche del *software* e alle domande poste dagli sviluppatori o da altre Amministrazioni per finalità di comprensione del funzionamento tecnico, riuso, collaborazione o sviluppo. L'Incaricato non è tenuto a rispondere ad altri soggetti o fornire assistenza sull'utilizzo del *software* o dare risposte sull'uso che l'Amministrazione fa del *software* o in generale su altri argomenti di competenza dell'Amministrazione.

### Contributi di codice

I contributi di codice inviati attraverso i meccanismi di collaborazione previsti dalla piattaforma di code hosting scelta (ad es. attraverso una *pull request*) dovranno essere valutati dall'Incaricato (MUST) che provvederà a dare un riscontro all'utente con considerazioni sulla fattibilità dell'integrazione (MUST). L'Incaricato è tenuto ad incorporare tutti i contributi di codice (SHOULD) che non presentano incompatibilità con gli obiettivi della fornitura, fornendo al contributore adeguata spiegazione in caso di diniego.





---

### Allegato C: Guida alle licenze Open Source

---

Le licenze Open Source possono essere molteplici, con lievi differenze che possono presentare, al momento del riuso del software, incompatibilità importanti. Questa guida ha lo scopo di fornire al lettore una breve introduzione alle diverse licenze adottabili, e suggerire l'adozione di alcune licenze specifiche. Limitare il numero di licenze in uso nel parco *software* ha infatti il vantaggio di semplificare di molto l'integrazione, dunque permettendo un risparmio da parte della Pubblica Amministrazione.

#### C.1 Versionamento delle licenze

Ogni licenza riportata di seguito citata dispone di un numero di versione, che garantisce agli enti redattori la possibilità di aggiornarla. Ai fini di garantire una compatibilità e riutilizzabilità del codice in futuro, in particolare per le licenze copyleft, si consiglia rilasciare ogni *software* secondo l'ultima versione disponibile della licenza, esplicitando la compatibilità con qualunque modificazione successiva.

#### C.2 Public Domain

Una licenza di pubblico dominio è una licenza nella quale il detentore dei diritti rinuncia ai diritti di proprietà intellettuale.

È la licenza consigliata per il rilascio di banche dati in Open Data, ove non ci si occupi di banche dati originali per scelta o disposizione di opere. Si ricorda che i diritti sui singoli elementi contenuti in una banca dati sono comunque coperti da una loro licenza individuale, che non è pregiudicata dal loro inserimento in una banca dati.

Questa licenza infatti offre ai riutilizzatori una flessibilità totale e riduce le complicazioni collegate all'operatività su varie e diverse licenze con il potenziale conflitto di disposizioni che comporta.

La più diffusa licenza in questo senso è la Creative Commons Zero (Codice SPDX: CC0-1.0).

#### C.3 Licenze non-copyleft

Le licenze non-copyleft sono licenze aperte che garantiscono molta libertà e flessibilità di riutilizzo da parte degli utenti.

Con queste licenze il detentore dei diritti richiede infatti solamente una frase che identifica la fonte del documento e, ove fattibile, un collegamento alle pertinenti informazioni sulla licenza, non ponendo limitazioni all'utilizzo o alla modifica dell'opera di ingegno.

Queste licenze vengono utilizzate per componenti *software* che implementano adattatori o componenti che nascono per essere integrati in applicativi di terze parti. Il focus principale di queste licenze è posto sul riutilizzo nel maggior numero di programmi possibili.

A differenza delle licenze copyleft, non esiste un obbligo di chi adatta questi componenti a rilasciare eventuali modifiche e miglioramenti, ma solo un riferimento per ottenere copia del codice sorgente originale.

Esempi di queste licenze sono la licenza BSD, la licenza MIT e la licenza Apache - codici SPDX: BSD-3-Clause, MIT e Apache-2.0.

**Nota:** l'utilizzo della licenza Apache è sconsigliato in quanto incompatibile con la licenza GNU GPL versione 2 (codice SPDX: GPL-2.0-or-later).

### C.4 Licenze copyleft

Le licenze copyleft sono licenze che richiedono un'attribuzione all'autore originale ma che aggiungono le cosiddette clausole «virali». Il concetto di viralità impone a ogni successiva modifica di essere rilasciata sotto una licenza che non imponga ulteriori restrizioni all'utente.

Si utilizzano per preservare la libertà del *software* a ogni successiva modifica, imponendo il rilascio in riuso di ogni eventuale modifica migliorativa da parte di terze parti.

Si noti che è indispensabile fornire copia del codice sorgente solo all'atto della distribuzione o concessione in licenza del software, non al momento dello sviluppo.

Questo tipo di licenza è consigliato per tutti gli applicativi software completi.

La licenza più diffusa in questo senso è la GNU GPL (codice SPDX: GPL-3.0-or-later) o la GNU AGPL, sua modificazione che copre anche l'ambito di *software* distribuiti via rete (codice SPDX: AGPL-3.0-or-later).

### C.5 Licenze copyleft - per le librerie

Ai fini di garantire flessibilità nel riutilizzo, ovvero permettere l'utilizzo di un componente *software* in un applicativo sotto qualunque licenza, la clausola di viralità può essere indebolita. Queste licenze vengono anche dette «a copyleft debole».

Questa clausola aggiuntiva mantiene intatta la viralità per quanto riguarda modifiche sul codice del componente, ma permette l'integrazione esterna da parte di un *software* distribuito sotto qualunque licenza.

Si noti che queste licenze contengono la cosiddetta «clausola di riproducibilità»: deve essere sempre possibile per l'utente di un software di terze parti sostituire il componente rilasciato sotto licenza LGPL con una sua versione modificata, senza modificare le funzionalità del *software* di terze parti. Questa clausola può essere problematica e non soddisfabile in alcuni ambienti embedded, ad esempio iOS.

La versione più utilizzata è la licenza GNU LGPL (codice SPDX: LGPL-3.0-or-later), versione modificata della licenza GNU GPL.

### C.6 Licenze non comprese nella classificazione appena introdotta

- La Mozilla Public License (codice SPDX: MPL-2.0) è una licenza copyleft, equiparabile alla licenza GNU GPL, che garantisce la libera distribuzione del codice ma impone di non poter riutilizzare, all'occorrenza di modifiche, loghi, nomi, o altri marchi registrati del detentore dei diritti, salvo ulteriore autorizzazione.

- La European Union Public License (EURL) è una licenza copyleft di tipo debole, sviluppata dalla Commissione europea, e tradotta ufficialmente in tutte le lingue dell'Unione. Questa licenza allega una tabella di compatibilità con alcune delle licenze aperte più comuni. Codice SPDX: `EURL-1.2`.

## C.7 Le licenze Creative Commons

Esistono molte versioni delle licenze Creative Commons. Queste licenze, così come suggerito da Creative Commons stesso, non sono utilizzabili per proteggere software, ma solo altre opere di ingegno (ad esempio documentazione o testo).

Le uniche licenze Creative Commons che possono intendersi licenze aperte, secondo quanto descritto in 3.3, sono:

- Creative Commons Zero - dominio pubblico (codice SPDX: `CC0-1.0`).
- Creative Commons Attribution (versione 4 e successive) - una licenza non copyleft (codice SPDX: `CC-BY-4.0`).
- Creative Commons Attribution-Share Alike (versione 4 e successive) - una licenza copyleft (codice SPDX: `CC-BY-SA-4.0`).

## C.8 Licenza applicabile alla documentazione e allegati del software

Tutti gli allegati al puro codice sorgente del *software* quali commenti nel codice sorgente, documentazione, esempi, schermate dimostrative, video, ecc. si considerano inclusi sotto la stessa licenza del software stesso. In generale, non è quindi necessario determinare licenze diverse per questi contenuti, se rilasciate contestualmente al *software* e parte integrante di esso.

Nel caso di rilascio di documentazione a sé stante rispetto al software, o nel caso questa sia particolarmente corposa (oltre una decina di pagine stampate), si consiglia di attribuire comunque una licenza all'opera. Si rimanda a 4.4 per una guida nella scelta della licenza migliore.

## C.9 Compatibilità tra le licenze

La compatibilità delle licenze dipende dalla cessione dei diritti intellettuali da parte dell'autore. Le licenze che in questo senso cedono meno diritti, al fine di preservare maggiormente nel tempo la libertà e riutilizzabilità del *software* creato, sono le licenze copyleft.

Quando si parla di compatibilità occorre distinguere due casi:

- La creazione di una nuova opera a partire da componenti già esistenti, con licenza unica
- L'assemblaggio e la distribuzione di più componenti interagenti, ognuna con licenza differente.

Per quanto riguarda il caso di creazione di una nuova opera sotto una licenza unica, la matrice di compatibilità è la seguente:

- Opere rilasciate sotto dominio pubblico sono rilasciabili con qualunque altra licenza
- Opere rilasciate sotto licenze non-copyleft sono rilasciabili con licenze copyleft
- Opere rilasciate sotto licenze copyleft possono essere solo rilasciate con licenze copyleft, a condizione che le due licenze siano compatibili

Nel secondo caso invece:

- Opere rilasciate sotto licenza di pubblico dominio, non-copyleft o copyleft debole possono interagire come componenti a sé stanti con qualunque altro applicativo, pur rispettando le eventuali clausole riguardo riferimenti al codice originali e la distribuzione di eventuali modifiche.

- Opere rilasciate sotto licenza copyleft possono interagire come componenti a sé stanti solo con altri componenti rilasciati con licenza copyleft compatibile.

---

### Allegato D: Guida alla presa in riuso di *software* open source

---

Questa guida è rivolta alle Amministrazioni che vogliono prendere a riuso un *software* o adottare un *software* open source di terzi, effettuando delle modifiche allo stesso. La guida può essere utilizzata da chiunque sia incaricato di svolgere le attività in essa descritte: le risorse interne dell'Amministrazione, società in-house dell'Amministrazione, un fornitore di servizi individuato dall'Amministrazione. Nel corso della descrizione delle attività si utilizza il termine "Incaricato" indistintamente per tutte e tre queste categorie.

La guida è stata prodotta anche per poter essere allegata ad un capitolato tecnico nell'ambito di un appalto; in questo caso l'Incaricato è tenuto a svolgere le attività descritte nel presente documento come parte integrante dell'appalto, in aggiunta a quanto specificato nel resto del capitolato.

Nel documento sarà adottata la seguente convenzione:

- **MUST/MUST NOT**: prescrizioni obbligatorie che l'Incaricato è tenuto a rispettare;
- **SHOULD/SHOULD NOT**: raccomandazioni che l'Incaricato è tenuto a valutare ed attuare qualora non vi siano documentabili ragioni ostative;
- **MAY/MAY NOT**: scelte che l'Incaricato può attuare a propria discrezione.

#### D.1 Modifica di *software* open source adottato in riuso

Nel caso di riuso di il *software* open source, la cui titolarità sia di una Pubblica Amministrazione, si applica quanto disposto da *Allegato A: Guida alla pubblicazione di software come open source*, unitamente alle prescrizioni contenute nella presente guida.

Inoltre, le procedure descritte in questa guida possono essere applicate anche alle modifiche effettuate a componenti *software* distribuiti con licenza open source la cui titolarità non sia delle Pubbliche Amministrazioni, di cui sia necessaria l'integrazione in *software* di proprietà della Pubblica Amministrazione.

In caso di adozione di *software* rilasciato da Pubblica Amministrazione, è necessario notificare l'adozione in riuso tramite l'apertura di un ticket (o analogo meccanismo quale una *pull request*) nel repository della Pubblica Amministrazione titolare, così che questa possa indicare i riferimenti al riuso all'interno del file *publiccode.yml* nella apposita sezione.

## D.2 Modifica del codice sorgente

L'Incaricato deve (MUST) operare con attenzione al fine di minimizzare il grado di divergenza tra il codice sorgente originale e quello modificato risultante dal lavoro effettuato. Nell'operare le modifiche necessarie all'adattamento non va infatti tenuto conto solo delle funzionalità richieste ma bisogna tendere a mantenere compatta e unitaria la base di codice.

La modifica del codice sorgente deve (MUST) essere ridotta al minimo indispensabile, preferendo invece i seguenti interventi:

- laddove il *software* originale preveda un meccanismo di plugin le nuove funzionalità dovranno (MUST) essere sviluppate sotto forma di plugin senza modificare il *core* (ad esempio, nel caso di un Content Management System);
- laddove sia possibile estendere le classi o in generale moduli esistenti senza modificarne il codice (cioè per *aggiunta*, sfruttando punti di estensione esistenti), è necessario (MUST) seguire questa strada.

Qualora non sia possibile realizzare tutte le funzionalità mediante i sopra descritti meccanismi di estensione, ma sia necessario modificare il codice sorgente originario, le modifiche devono essere ispirate al minimalismo, ovvero in ordine di preferenza:

- si deve implementare solo quanto necessario a poter operare secondo una delle modalità di estensione sopra descritte;
- si devono implementare le nuove funzionalità non nell'ottica di specializzare il *software* originario al proprio contesto, ma al contrario concependole come un intervento di potenziamento e generalizzazione del *software* originario.

Non sono ammessi interventi di modifica che limitino le funzionalità o i casi di uso del *software* originario (MUST NOT); ogni intervento deve essere un miglioramento e deve essere concepito in modo che possa essere recepito come contributo da parte dei maintainer del *software* originale (SHOULD).

In ogni caso, nel README dovrà essere chiaramente spiegato cosa è stato modificato rispetto al progetto originale.

Il repository pubblicato dovrebbe (SHOULD) contenere tutta la storia delle modifiche dei "code commit" che l'Incaricato ha effettuato durante il processo di sviluppo, preservando lo storico dell'operato dell'attività di sviluppo, necessario e utile a tutti gli sviluppatori che vorranno contribuire per ridurre la curva di apprendimento.

## D.3 Interazione con il maintainer del progetto originale

All'Incaricato è richiesto di massimizzare l'interazione con il *maintainer* del progetto originale (SHOULD), con approccio collaborativo e con l'obiettivo di consolidare il lavoro in una unica base di codice a beneficio del successivo riuso.

Nel caso di correzioni di bug, l'Incaricato è tenuto (MUST) ad inviare al *maintainer* originale la proposta di correzione usando gli strumenti di collaborazione previsti dalla piattaforma di code hosting (ad es. *pull request*).

Nel caso di modifiche necessarie per implementare le nuove funzionalità, l'Incaricato è tenuto (MUST) a prendere contatto con il *maintainer* attraverso i canali pubblici del repository (*issue tracker*) in modo da presentare il nuovo caso d'uso, proporre la modifica ed ottenere feedback sulle modalità da seguire soprattutto nell'ottica di scrivere modifiche che possano essere incorporate dal *maintainer* originale. È necessario concedere alcuni giorni al *maintainer* per rispondere; tuttavia, qualora i tempi di risposta si dovessero protrarre eccessivamente, si potrà procedere anche in autonomia (MAY).

Al termine dello sviluppo, l'Incaricato è tenuto a proporre al *maintainer* originale le proprie modifiche (MUST), con delle proposte di codice (*pull request*) granulari, ovvero distinte per singole funzionalità in modo da consentire al *maintainer* di valutarle singolarmente.

l'Incaricato è inoltre tenuto (MUST) a tenere traccia di tutte le contribuzioni al *software* inviate al *maintainer* del *software* originale, documentandone lo stato di integrazione all'interno del file README del repository.

## D.4 Pubblicazione di codice open source non originato nel originato nel contesto della PA

Nel caso in cui il *maintainer* di un *software* open source la cui titolarità non sia attribuita ad una Pubblica Amministrazione, il cui abbia recepito integralmente le proposte di modifica (v. paragrafo precedente) presentate del l'Incaricato, quest'ultimo è comunque tenuto a pubblicare il codice nello strumento di code hosting dell'Amministrazione per metterlo a riuso, specificando nel *README* che tale codice è stato recepito dal progetto originario, con un *link* al *repository* dello stesso.

Come prescritto dalle Linee Guida, il «software a riuso» è il software rilasciato da una Pubblica Amministrazione in adempimento all'art 69 del CAD. Quindi l'Amministrazione Pubblica che adotta un *software* open source non originato nel contesto della PA, è tenuta a metterlo a riuso, indicando la sua provenienza.





---

### Allegato D: Guida alla presa in riuso di *software* open source

---

Questa guida è rivolta alle Amministrazioni che vogliono prendere a riuso un *software* o adottare un *software* open source di terzi, effettuando delle modifiche allo stesso. La guida può essere utilizzata da chiunque sia incaricato di svolgere le attività in essa descritte: le risorse interne dell'Amministrazione, società in-house dell'Amministrazione, un fornitore di servizi individuato dall'Amministrazione. Nel corso della descrizione delle attività si utilizza il termine "Incaricato" indistintamente per tutte e tre queste categorie.

La guida è stata prodotta anche per poter essere allegata ad un capitolato tecnico nell'ambito di un appalto; in questo caso l'Incaricato è tenuto a svolgere le attività descritte nel presente documento come parte integrante dell'appalto, in aggiunta a quanto specificato nel resto del capitolato.

Nel documento sarà adottata la seguente convenzione:

- **MUST/MUST NOT**: prescrizioni obbligatorie che l'Incaricato è tenuto a rispettare;
- **SHOULD/SHOULD NOT**: raccomandazioni che l'Incaricato è tenuto a valutare ed attuare qualora non vi siano documentabili ragioni ostative;
- **MAY/MAY NOT**: scelte che l'Incaricato può attuare a propria discrezione.

#### E.1 Modifica di *software* open source adottato in riuso

Nel caso di riuso di il *software* open source, la cui titolarità sia di una Pubblica Amministrazione, si applica quanto disposto da *Allegato A: Guida alla pubblicazione di software come open source*, unitamente alle prescrizioni contenute nella presente guida.

Inoltre, le procedure descritte in questa guida possono essere applicate anche alle modifiche effettuate a componenti *software* distribuiti con licenza open source la cui titolarità non sia delle Pubbliche Amministrazioni, di cui sia necessaria l'integrazione in *software* di proprietà della Pubblica Amministrazione.

In caso di adozione di *software* rilasciato da Pubblica Amministrazione, è necessario notificare l'adozione in riuso tramite l'apertura di un ticket (o analogo meccanismo quale una *pull request*) nel repository della Pubblica Amministrazione titolare, così che questa possa indicare i riferimenti al riuso all'interno del file *publiccode.yml* nella apposita sezione.

## E.2 Modifica del codice sorgente

L'Incaricato deve (MUST) operare con attenzione al fine di minimizzare il grado di divergenza tra il codice sorgente originale e quello modificato risultante dal lavoro effettuato. Nell'operare le modifiche necessarie all'adattamento non va infatti tenuto conto solo delle funzionalità richieste ma bisogna tendere a mantenere compatta e unitaria la base di codice.

La modifica del codice sorgente deve (MUST) essere ridotta al minimo indispensabile, preferendo invece i seguenti interventi:

- laddove il *software* originale preveda un meccanismo di plugin le nuove funzionalità dovranno (MUST) essere sviluppate sotto forma di plugin senza modificare il *core* (ad esempio, nel caso di un Content Management System);
- laddove sia possibile estendere le classi o in generale moduli esistenti senza modificarne il codice (cioè per *aggiunta*, sfruttando punti di estensione esistenti), è necessario (MUST) seguire questa strada.

Qualora non sia possibile realizzare tutte le funzionalità mediante i sopra descritti meccanismi di estensione, ma sia necessario modificare il codice sorgente originario, le modifiche devono essere ispirate al minimalismo, ovvero in ordine di preferenza:

- si deve implementare solo quanto necessario a poter operare secondo una delle modalità di estensione sopra descritte;
- si devono implementare le nuove funzionalità non nell'ottica di specializzare il *software* originario al proprio contesto, ma al contrario concependole come un intervento di potenziamento e generalizzazione del *software* originario.

Non sono ammessi interventi di modifica che limitino le funzionalità o i casi di uso del *software* originario (MUST NOT); ogni intervento deve essere un miglioramento e deve essere concepito in modo che possa essere recepito come contributo da parte dei maintainer del *software* originale (SHOULD).

In ogni caso, nel README dovrà essere chiaramente spiegato cosa è stato modificato rispetto al progetto originale.

Il repository pubblicato dovrebbe (SHOULD) contenere tutta la storia delle modifiche dei "code commit" che l'Incaricato ha effettuato durante il processo di sviluppo, preservando lo storico dell'operato dell'attività di sviluppo, necessario e utile a tutti gli sviluppatori che vorranno contribuire per ridurre la curva di apprendimento.

## E.3 Interazione con il maintainer del progetto originale

All'Incaricato è richiesto di massimizzare l'interazione con il *maintainer* del progetto originale (SHOULD), con approccio collaborativo e con l'obiettivo di consolidare il lavoro in una unica base di codice a beneficio del successivo riuso.

Nel caso di correzioni di bug, l'Incaricato è tenuto (MUST) ad inviare al *maintainer* originale la proposta di correzione usando gli strumenti di collaborazione previsti dalla piattaforma di code hosting (ad es. *pull request*).

Nel caso di modifiche necessarie per implementare le nuove funzionalità, l'Incaricato è tenuto (MUST) a prendere contatto con il *maintainer* attraverso i canali pubblici del repository (*issue tracker*) in modo da presentare il nuovo caso d'uso, proporre la modifica ed ottenere feedback sulle modalità da seguire soprattutto nell'ottica di scrivere modifiche che possano essere incorporate dal *maintainer* originale. È necessario concedere alcuni giorni al *maintainer* per rispondere; tuttavia, qualora i tempi di risposta si dovessero protrarre eccessivamente, si potrà procedere anche in autonomia (MAY).

Al termine dello sviluppo, l'Incaricato è tenuto a proporre al *maintainer* originale le proprie modifiche (MUST), con delle proposte di codice (*pull request*) granulari, ovvero distinte per singole funzionalità in modo da consentire al *maintainer* di valutarle singolarmente.

l'Incaricato è inoltre tenuto (MUST) a tenere traccia di tutte le contribuzioni al *software* inviate al *maintainer* del *software* originale, documentandone lo stato di integrazione all'interno del file README del repository.

## E.4 Pubblicazione di codice open source non originato nel originato nel contesto della PA

Nel caso in cui il *maintainer* di un *software* open source la cui titolarità non sia attribuita ad una Pubblica Amministrazione, il cui abbia recepito integralmente le proposte di modifica (v. paragrafo precedente) presentate del l'Incaricato, quest'ultimo è comunque tenuto a pubblicare il codice nello strumento di code hosting dell'Amministrazione per metterlo a riuso, specificando nel *README* che tale codice è stato recepito dal progetto originario, con un *link* al *repository* dello stesso.

Come prescritto dalle Linee Guida, il «software a riuso» è il software rilasciato da una Pubblica Amministrazione in adempimento all'art 69 del CAD. Quindi l'Amministrazione Pubblica che adotta un *software* open source non originato nel contesto della PA, è tenuta a metterlo a riuso, indicando la sua provenienza.



## APPENDICE F

---

### Allegato E: Tabella Sinottica degli elementi necessari al percorso decisionale

---

Al fine di agevolare la valutazione comparativa, attraverso un percorso decisionale per le PP.AA., che tenga conto delle indicazioni riportate sia nell'articolo 68 che nel 69 del CAD, si fa riferimento al quadro sinottico che segue:

**Linee Guida su acquisizione e riuso di software per le pubbliche amministrazioni, Release Finale**

Soluzioni Art. 68	Obbligo di mettere a riuso Art. 69 comma 1	Obbligo di acquisire la titolarità Art. 69 comma 2	Obbligo Valutazione economica (TCO) Art. 68 comma 1bis	Obbligo Valutazione tecnica Art. 68 comma 1ter	Assicurare l'interoperabilità tra PA Art. 68 comma 1bis	Garanzie sulla sicurezza Art. 68 comma 1bis	Conformità normativa in materia di privacy Art. 68 comma 1bis	Livelli di servizio adeguati Art. 68 comma 1bis
Software sviluppato per conto della pubblica amministrazione	Sì	Sì	Sì, con l'esclusione dell'acquisto	Sì	Sì	Sì	Sì	Sì
Riutilizzo di software o parti di esso sviluppati per conto della pubblica amministrazione	Sì, solo in caso di modifica	Sì	Sì, con l'esclusione dell'acquisto	Sì	Sì	Sì	Sì	Sì
Software libero o a codice sorgente aperto	Sì, solo in caso di modifica	No	Sì, con l'esclusione dell'acquisto	Sì	Sì	Sì	Sì	Sì
Software fruibile in modalità cloud computing	Sì, solo per il software già di proprietà o implementato ad hoc per la PA	Sì, solo per il software già di proprietà o implementato ad hoc per la PA	Sì	Sì	Sì	Sì	Sì	Sì
Software di tipo proprietario mediante ricorso a licenza d'uso	No, ad eccezione del software creato per rendere possibile l'interoperabilità applicativa (ad esempio API)	No, ad eccezione del software creato per rendere possibile l'interoperabilità applicativa (ad esempio API)	Sì	Sì	Sì	Sì	Sì	Sì
Software combinazione delle precedenti soluzioni	Sì, solo per il software già di proprietà o implementato ad hoc per la PA	Sì, solo per il software già di proprietà o implementato ad hoc per la PA	Sì	Sì	Sì	Sì	Sì	Sì