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

Questa guida è rivolta alle Amministrazioni che, titolari di un software, lo vogliano 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>