Em002-4 Guida Community OSS

Raccomandazione sull'informatica nell'Amministrazione federale[1]

Il presente documento è un allegato indipendente del documento principale Em002.

Classificazione:[2]	Non classificato
Carattere vincolante:[3]	Raccomandazione
Settore di pianificazione:[4]	TIC dell'Amministrazione federale
Versione attuale:	2.0
Sostituisce versione:	1.0 del 24.02.2025
Stato:	Approvato
Data di approvazione (versione attuale):	9.12.2025 (Version italiana del 15.4.2026)
Approvato da, base legale:	Il Delegato alla trasformazione digitale e la governance delle TIC (D-TDT), ai sensi dell'articolo 40 dell'Ordinanza del 1° maggio 2025 sui Servizi Digitali e la Trasformazione Digitale nell'Amministrazione Federale (Ordinanza sulla Digitalizzazione, DigiV), SR 172.019.1
Lingue:	Tedesco (originale), francese e italiano (traduzioni)
Licenza	CC0 1.0 Universal Questo documento è pubblicato sotto la licenza CC0 e può quindi essere utilizzato, modificato e distribuito in ogni formato e anche a scopi commerciali.

1 «Empfehlung zur Bundesinformatik» secondo [P035], n. 4.6

2 Riguardo alla classificazione AD USO INTERNO e CONFIDENZIALE cfr. ordinanza dell'8 novembre 2023 sulla sicurezza delle informazioni (OSIn, RS **128.1**)

3 Si veda la nota 1.

4 Settori di pianificazione secondo la Strategia TIC della Confederazione 2020–2023 del 3 aprile 2020 (SB000)

Indice

1 L’essenziale in breve

Qui di seguito sono esposti i punti centrali del documento.

- La creazione o la gestione di una community OSS **non è richiesta** ai sensi dell’articolo 9 LMeCA.
- Una community è rilevante se si pensa che possa risultare utile un gruppo di utenti e si vogliono creare **sinergie** con altri utenti del software.
- Una community non si limita al software e può comprendere anche i processi nell’ambiente circostante e la standardizzazione dei dati.
- È sempre necessario un **impegno iniziale** per creare una community, ed è solitamente di lunga durata. In molti casi è opportuno che l’ufficio responsabile si faccia anche carico dei costi di coordinamento.
- Una community deve apportare dei **benefici**; servono quindi interessati e partecipanti attivi.
- Una community può anche instaurarsi autonomamente ed essere formalizzata solo in seguito.
- L’obiettivo di una community è aumentare i **benefici a lungo termine per tutti** i suoi membri, secondo il motto «guadagno almeno tanto quanto ho investito».
- Una community deve essere **strutturata nel modo più semplice possibile**.
- Nella fase iniziale, bisogna verificare brevemente la community utilizzando la lista di controllo [Em002-4.1]. Si continuerà a svilupparla solo se questa risulta effettivamente utile.
- La formalizzazione avviene per mezzo di un **piano per la community**, che si deve basare il più possibile su fonti esistenti. Motto: «Non inventare l’acqua calda».
- Se si intende avviare una collaborazione, occorre tenerne conto sin dall’inizio del progetto, poiché i potenziali partner dovrebbero essere coinvolti già nella fase di raccolta dei requisiti. È inoltre necessario pianificare fin da subito e in modo accurato gli aspetti giuridici e relativi al diritto in materia di appalti pubblici.
- Prima di aderire a una collaborazione esistente, è necessario verificare le possibilità giuridiche e in materia di appalti pubblici.
- Per i contributi non è necessario creare una community. Se però questo è richiesto dal progetto, bisogna firmare il relativo «**contributor license agreement»** (**CLA**) oppure gli sviluppatori devono essere autorizzati dal progetto a fornire il software con un «developer certificate of origin» (DCO).

2 Introduzione

La presente guida è pensata per gli specialisti TIC responsabili di un'applicazione e che vogliono offrirla come open source oppure che collaborano a un'applicazione di questo tipo. Queste attività devono essere organizzate in modo formale o informale. Il presente documento contiene informazioni importanti sull'organizzazione e la creazione di una community[5] e anche sullo sviluppo aperto direttamente come progetto open source.

Come secondo elemento per favorire la standardizzazione, all'interno del presente documento viene riportato un elenco di piani per community già esistenti. L'idea è quella di poter costituire nuove community con il minimo sforzo basandosi su altre soluzioni ben riuscite.

Il presente documento non contiene raccomandazioni esplicite per l'organizzazione di community di librerie software (parti di un software che adempiono funzioni parziali, p. es. la generazione di PDF). Per queste ultime, in genere, non si elabora un piano a parte, ma si utilizza il modello semplice definito nel template di archiviazione dei repository.

Il documento «Em002-1 Guida pratica Software a codice sorgente aperto nell'Amministrazione federale» presenta al numero 4 forme di collaborazione basilari, al numero 8 i diversi modelli di supporto e nell'appendice indicazioni sul comportamento del mercato degli open source.

[5] Tipicamente, la community di un progetto open source rappresenta una piramide.
Alla base vi sono gli utenti del software, in particolare quelli impegnati che si rendono attivamente partecipi della community, per esempio compilando «bug report» o richiedendo caratteristiche specifiche oppure ancora inserendo contributi nelle mailing list.
Subito sopra di loro nella piramide si trovano i «contributor». Si tratta di membri della community che propongono i propri contributi al codice e in genere non hanno diritti di scrittura sul «repository». I loro contributi vengono esaminati dai «maintainer» del progetto e adottati nel «repository» una volta raggiunta la qualità tipica del progetto.
In cima alla piramide si trovano i «maintainer» o i «committer» di un progetto. In questo ruolo, uno sviluppatore ha una maggiore responsabilità, che, per esempio, si rende evidente dal fatto che può accettare i contributi. Questo diritto si esprime spesso attraverso i diritti di scrittura sul «repository». A questo livello avviene il controllo sul software, sulla sua qualità e sulla varietà delle sue funzioni. In progetti più complessi, questo livello può essere ulteriormente ampliato. È noto per esempio il kernel di Linux in cui sono presenti «maintainer» di sottosistema a diversi livelli e in cui alla fine un solo sviluppatore, Linus Torvalds, inserisce le patch nel «repository» del progetto. Un ulteriore esempio sono i progetti della «Eclipse Foundation», in cui tipicamente è presente anche una direzione di progetto che detiene diritti aggiuntivi nel contesto del processo di sviluppo della fondazione stessa, la quale in questo modo può per esempio attivare il processo di release o avviare la scelta formale dei «committer» o della direzione di progetto [BITKOM2023].

3 Tipologia e scopo di una community

Una community può perseguire uno dei seguenti obiettivi:

- Raccolta di ulteriori requisiti
- Diffusione delle conoscenze relative all'applicazione
- Migliore feedback da parte degli utenti
- Promozione di traduzioni, documentazione e formazione
- Diffusione della standardizzazione
- Miglioramento generale dell'interazione con gli utenti
- Ampliamento della collaborazione ad altre parti
- Promozione di ulteriori software basati sul software

Il tipo di community più adatto a un'applicazione dipende fortemente dall'ambito specialistico, dai potenziali utenti nonché dalla strategia dell'istituzione che si occupa della pubblicazione. Esistono moltissime varianti di community. Idealmente, se il progetto è adatto, si può anche aderire a una community esistente. In alternativa, se ne può creare una con uno o più partner paritari.

È importante che si stabilisca **una governance** e che i punti rilevanti siano fissati in un **piano per la community** (cfr. anche [BITKOM2023] n. 4.1 e [IZqCab2023].

È bene notare che le community dedicate ai software possono essere combinate anche con quelle per i dati, la standardizzazione e la tematica aziendale. La community può così comprendere processi, standardizzazione, software e dati. In determinate circostanze si possono utilizzare anche contenitori come «eOperations», ADS e eCH.

Un aspetto importante: una community funziona se i partecipanti ricevono più di quanto investono.Possono verificarsi i seguenti scenari:

- **Community per un OSS della Confederazione**: è necessario elaborare un piano per la governance e la community. Inoltre, occorre anche regolamentare le modalità con cui i terzi possono fornire contributi.
- **Nuova collaborazione per la risoluzione di un problema**: oltre al piano per la governance e la community, si deve creare l'intera struttura giuridica per la collaborazione.
- **Partecipazione a una collaborazione**: l'Amministrazione federale deve verificare se e come questo sia possibile.
- **Contributo a un progetto di terzi**: va verificata la policy del progetto per quanto riguarda i contributi (CLA, DCO ecc.).

Le questioni giuridiche relative all'acquisizione sono trattate nel documento «Em002-7 Aspetti strategici relativi agli appalti e ai software open source».

3.1 Caso particolare: nuova collaborazione per la creazione, la manutenzione e il supporto di OSS

I costi complessivi per un OSS diminuiscono se la creazione e l'ulteriore sviluppo avvengono nell'ambito di una collaborazione. Ciò può avvenire anche senza cooperazione diretta o semplicemente con una pianificazione congiunta.

Se l'obiettivo è una collaborazione più formale, occorre verificare il mercato per assicurarsi che una collaborazione sia l'opzione più sensata. La redditività delle collaborazioni è

maggiore rispetto a quella ottenibile se ciascuno sviluppasse autonomamente i propri prodotti (cfr. «Em002-4.1 Lista di controllo Community OSS».

Si noti che è opportuno avviare la collaborazione molto presto nel processo HERMES[6], più precisamente già durante l'analisi della situazione e la definizione dei requisiti.

In linea di principio, la collaborazione in Svizzera è permessa ai sensi dell'articolo 4 LMeCA.

Una collaborazione ha anche una componente giuridica e relativa al diritto in materia di appalti pubblici. Al momento si tratta sempre di casi singoli, ma con il tempo si affermeranno dei modelli standardizzati.

3.1.1 Componenti giuridici

È possibile iscriversi a delle associazioni. In genere bisogna valutare caso per caso, considerando chi c'è dietro e quali impegni vengono assunti. È possibile delegare l'iscrizione a eOperations o organizzazioni simili.

Fintanto che la collaborazione non comporta costi o l'assunzione di impegni formali, è possibile stipulare un memorandum d'intesa a livello di ufficio o di Cancelliere federale.

Bisogna evitare di stipulare un trattato internazionale. Inoltre, in due casi particolari[7] la collaborazione è già coperta:

- Contratti basati su un accordo internazionale relativo alla realizzazione congiunta di un progetto da parte degli Stati firmatari;
- Contratti che secondo procedure o condizioni particolari sono aggiudicati a un'organizzazione internazionale.

3.1.2 Componente relativa al diritto in materia di appalti pubblici

In molti casi la collaborazione avviene tra organizzazioni del settore pubblico. In questo caso l'acquisto non presenta problemi se si tratta di un «in-state» (cfr. il documento «Em002-7 Aspetti strategici relativi agli appalti e ai software open source»).

È possibile stipulare contratti con imprese estere senza sede in Svizzera, purché venga rispettato il diritto in materia di appalti pubblici.

L'acquisto delle risorse proprie avviene nell'ambito di un acquisto regolare o è già stato effettuato.

3.2 Caso particolare: adesione a una collaborazione esistente

In questo caso occorre stabilire una base giuridica e regolamentare gli aspetti relativi al diritto in materia di appalti pubblici. Il modo più semplice è partecipare alle spese secondo il principio «ognuno paga la propria parte». Possono essere comprese anche le spese generali, se vengono utilizzate risorse proprie o correttamente indicate.

Aderire a una collaborazione estera è più complicato. L'opzione più semplice è un'associazione tramite un memorandum d'intesa.

---

6 www.hermes.admin.ch/de/pjm-2022/verstehen/aufgaben/ausschreibung-erarbeiten.html

7 www.eda.admin.ch/deza/de/home/partnerschaften/auftraege-beitraege/auftraege/anforderungen/rechtlich.html

3.3 Caso particolare: contributo di software a progetti OSS di terzi

Se l'Amministrazione federale apporta modifiche al codice di un OSS esistente, è opportuno farlo direttamente «up-stream» nel progetto originale. In questo modo, la manutenzione non dovrà più essere effettuata dall'Amministrazione federale, fattore auspicabile ai sensi dell'articolo 9 LMeCA. Non sorgono problemi dal punto di vista del diritto in materia di appalti pubblici.

È importante che sia il titolare dei diritti a effettuare il contributo. Di conseguenza, l'Amministrazione federale si assume questo compito per i propri collaboratori e incaricati. Più precisamente, va firmato l'eventuale CLA del progetto o si deve allegare un DCO alla «pull request».

Nel caso dell'Amministrazione federale, un **DCO**[8] può presentarsi come segue:

Developer Certificate of Origin
Version 1.1

Copyright (C) 2004, 2006 The Linux Foundation and its contributors.

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Developer's Certificate of Origin 1.1

By making a contribution to this project, we certify that:

(a) The rights to this contribution are owned by the Swiss Confederation and I Am authorized to submit it under the open source license indicated in the file; or

(b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or

(c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.

(d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

La struttura di un CLA è invece descritta nel capitolo 8.9 del documento «Em002-3 Guida Licenze OSS».

Il trasferimento effettivo del codice avviene secondo le linee guida del progetto specifico.

[8] https://developercertificate.org/

4 Piano per la community

La struttura e la governance sono le componenti principali di una community e vengono esposte in un piano per la community.

Il piano per la community che deve essere redatto dal responsabile di progetto o dell'applicazione (o da ordinare presso terzi) deve rispettare la struttura illustrata di seguito. In base al tipo specifico di community, non tutti gli elementi sono necessari. Idealmente, in molti capitoli si rimanda a documenti già standardizzati o si utilizzano processi standard dei rispettivi «repository». Il piano per la community può essere utilizzato anche per progetti già esistenti e gestiti da altri, soprattutto nel caso in cui la governance effettiva di tali progetti non sia ancora stata stabilita per iscritto.

La formalizzazione della community raggruppa tutti i partecipanti e semplifica il processo di onboarding di nuovi partecipanti.

Bozza della struttura del piano per la community:

1. Tabella riassuntiva
   (Nome, «repository», indirizzo di contatto, licenza, livello di supporto, esistente/nuova)
2. Finalità
   a. Finalità dell'applicazione
   b. Finalità della community
3. Organizzazione
   a. Proprietario del codice
   b. Forma organizzativa / organismi
   c. Diritti di voto
   d. Gestione dei progetti
   e. Questioni legate all'acquisto (se del caso)
   f. Altri aspetti di governance
4. Roadmap e processo di modifica
5. Processo di sviluppo
   a. «Open development» (sviluppo aperto)
   b. Diritti dei «committer»
   c. Processo di revisione
   d. Distribuzione dei compiti e indicazione dei nomi
   e. Backup
6. Finanziamento della community
7. Commercializzazione
8. Gestione dei contributi esterni
   a. Gestione degli errori segnalati
   b. Gestione delle richieste
   c. Gestione delle «pull request»
   d. Gestione dei «fork»
   e. CLA, DCO e cessione di diritti/proprietà intellettuale
9. Funzionamento dell'applicazione

In generale, ogni autorità federale è responsabile della redazione dei propri piani per la community. Il settore TDT può pubblicare i modelli/esempi esistenti e riceve esempi corrispondenti.

È anche utile che i responsabili della comunità scambino informazioni tra di loro.Una possibile fonte per le parti è rappresentata anche dai documenti della «Eclipse Foundation»[9], anche se in questo caso sarebbe necessario personalizzare i contenuti secondo l'idea di «fare solo ciò che è necessario».

5 Decisioni di principio per la creazione di una community

I principi della community, o la loro mancanza, vengono individuati sulla base della «Em002-4.1 Lista di controllo Community OSS». Le questioni di principio rilevanti si evincono dallo schema morfologico nella Figura 1.

Figura 1: Schema morfologico community OSS

Per la concezione della community, si consiglia di **focalizzarsi soprattutto sul futuro prossimo**; se, per esempio, non vi sono ancora concretamente persone interessate all'applicazione, in genere è poco ragionevole definire una community con una società semplice, una propria associazione e processi complessi di elaborazione della roadmap[10][11].

5.1 Gestione del prodotto

A seconda del tipo di applicazione, della sua rilevanza strategica e del suo stato all'interno del ciclo di sviluppo, sono possibili diverse varianti per la gestione del prodotto.
Nelle organizzazioni interne alle autorità federali questo aspetto riguarda sempre i responsabili dell'applicazione. Per il piano per la community sono particolarmente rilevanti la definizione della roadmap specialistica e tecnica e i processi di release.

Possibilità:

a) **organizzazioni interne alla Confederazione**: la Confederazione intende mantenere il controllo e definisce da sola la roadmap;
b) **organizzazione congiunta**: insieme ad altri utenti o fornitori;
c) **organizzazione aperta**: relazioni informali, non è presente una roadmap attiva;
d) **terzi**: il prodotto esiste già oppure il lavoro deve essere affidato ad altri.

---

9 Cfr. www.eclipse.org/projects/handbook/#starting

10 Con un'associazione (o una fondazione) si crea una persona giuridica indipendente che si occupa del software e del prodotto. Questo dà risultati solo a partire da un certo livello di importanza, complessità e dall'esistenza di più partner (paritari). La roadmap serve a mostrare quanto pianificato a un pubblico più ampio e a potenziali utenti.

11 In determinate circostanze, proprio per quei progetti che fin dall'inizio comprendono diversi attori statali, è possibile rivolgersi a fondazioni esistenti e valutare se sia possibile gestire i progetti e la governance sotto il loro ombrello, analogamente a https://finos.org o https://osr.finos.org oppure www.eclipse.org/collaborations/ o https://iot.eclipse.org o https://outreach.eclipse.foundation/open-regulatory-compliance.

5.1.1 Organizzazioni interne alla Confederazione

Se l'applicazione ha un'elevata rilevanza strategica o se la Confederazione deve poter implementare rapidamente le proprie modifiche, allora può essere ragionevole mantenere il controllo. L'autorità federale decide da sola, tramite i responsabili dei requisiti, cosa si debba introdurre nel software e quando.

Ciò significa però che eventuali altri utenti dell'applicazione sono quasi costretti a utilizzare una propria versione («fork») dell'applicazione.

Diversamente, per loro sarebbe difficile attuare le modifiche e gli ampliamenti che ritengono importanti. Ciò non depone tuttavia contro una pubblicazione come open source e gli strumenti per la gestione del codice[12] offrono un ampio supporto per scenari simili; le modifiche e la correzione di errori possono essere intraprese selettivamente in entrambe le direzioni.

In questo modello, una ripartizione dei costi è in genere difficilmente possibile.

5.1.2 Organizzazione congiunta (la Confederazione dirige o è partner paritario)

Le decisioni relative alla roadmap e alle priorità devono essere concordate e prese insieme agli altri membri della community. In quanto membro della community, la Confederazione partecipa alla decisione ma non è l'organizzazione responsabile.

Perché l'iter decisionale congiunto possa funzionare si devono definire in anticipo strutture e regole. In questo contesto, si stabilisce anche come ripartire i costi di attuazione delle modifiche decise congiuntamente.

Con questo modello si genera un'unica versione dell'applicazione che viene quindi utilizzata da tutti i membri. Per questo motivo i costi complessivi sono molto ridotti rispetto alle altre varianti. D'altra parte, la flessibilità dei singoli utenti è però minore, in quanto essi devono allineare in anticipo i propri desideri di modifica a tutti gli altri; il processo decisionale è più lungo e dispendioso.

Questo modello si adatta meglio alle applicazioni che devono essere sviluppate insieme ad altre organizzazioni ben definite (p. es. i Cantoni).

5.1.3 Organizzazione aperta

In questo modello, i processi e le strutture sono definite solo superficialmente. La Confederazione mantiene il controllo a livello formale, ma è aperta a contributi e modifiche.

La roadmap è molto sommaria o non viene affatto definita, il consenso si ottiene in modo informale e in discussioni che avvengono direttamente sulla piattaforma di pubblicazione. La community stessa è molto dinamica, i membri possono essere molto attivi per un certo periodo di tempo e poi essere meno presenti.

Questo modello è adatto anche per applicazioni piuttosto piccole che possono già essere utilizzate in modo produttivo e sono «pronte».
È in genere la forma più adatta anche per le applicazioni o i componenti con orientamento tecnico che si rivolgono a un gruppo target potenzialmente più ampio e non esattamente definibile in partenza.

[12] Per esempio GitHub.

5.1.4 Terzi

In questo modello i processi e le strutture vengono definiti da una parte terza e la Confederazione è presente come membro della community ma non ha alcuna funzione direttiva. Questo può accadere se per esempio il progetto è stato definito da un altro Stato o Cantone. Oppure eventualmente la Confederazione stessa non è più interessata al proseguimento e cede ad altri il compito.

5.2 Coinvolgimento dei fornitori

Per quanto riguarda il coinvolgimento dei fornitori vi sono in sostanza le seguenti possibilità.

a) **Fornitore come mandatario**: in sostanza viene considerato come sostituibile nel medio termine. Il fornitore ha il compito di sviluppare ulteriormente l'applicazione mantenendo i costi efficienti e la qualità elevata in base agli incarichi assegnati dalla gestione del prodotto (o dalla community), ma sulla roadmap e sulla definizione delle priorità può avere al massimo un'influenza consultiva.
b) **Partner a lungo termine con diritto di codecisione**: i fornitori devono poter partecipare attivamente alle decisioni relative allo sviluppo. Questo rafforza il loro ruolo ed essi sono potenzialmente disposti a investire personalmente nell'applicazione. Di solito, in questo modello i fornitori vendono poi il software e cercano attivamente nuovi clienti.
c) **Indipendenti**: sono i fornitori stessi a definire se e come vogliono collaborare al progetto. Questa situazione può verificarsi, per esempio, se più fornitori vogliono generare affari basandosi sul software stesso. In determinate circostanze, le licenze liberali possono aiutare a raggiungere tale scopo e questo può funzionare anche se l'autorità federale e il fornitore hanno una strategia diversa.
   In questo scenario è probabile che il fornitore debba creare un «fork».
d) **Direzione**: se il fornitore prende a carico la gestione del prodotto, allora dirige il progetto e definisce quindi lo sviluppo. D'altra parte, questo può anche far sì che l'amministrazione debba assumersi pochissime responsabilità. Il fornitore può assumere questo ruolo di propria iniziativa nel caso di un rilascio minimo.

Se il fornitore assume un ruolo forte, questo va a vantaggio dell'amministrazione: la commercializzazione attiva dell'applicazione può contribuire a una crescita più rapida e intensa della community e quindi anche far calare più velocemente i costi per membro.

Se si sceglie un modello con forte codeterminazione dei fornitori si deve prestare attenzione affinché questo non provochi scostamenti dalle normative prescritte in ambito di diritto in materia di appalti pubblici. In generale è raccomandabile coinvolgere almeno due fornitori in modo che non si crei una dipendenza eccessiva da un unico fornitore. Inoltre, dovrebbero avere la possibilità di accedere alla community anche altri fornitori.

5.3 Ripartizione dei costi

Per la suddivisione dei costi di manutenzione, ulteriore sviluppo e nuove funzioni, esistono in genere le seguenti varianti:

a) **Organizzazioni interne alla Confederazione**: soprattutto se l'autorità federale intende creare un ecosistema o se vuole mantenere interamente il controllo, deve anche assumersi i costi.
b) **Ognuno per sé**: ognuno paga per le modifiche che desidera implementare. Eventualmente si può decidere caso per caso se pagare congiuntamente determinati

ampliamenti.
Per i costi ricorrenti è presente inoltre una chiave di ripartizione oppure il conteggio viene svolto in base al fabbisogno.

c) **Principio di causalità**: chi necessita di prestazioni di servizi e caratteristiche se ne assume anche i costi. Eventualmente si possono applicare tariffe orarie standardizzate.
d) **Chiave di ripartizione**: i costi vengono suddivisi in base a una chiave definita all'interno della community. Sono previste deroghe solo per ampliamenti specifici che devono essere pagati direttamente dagli utenti interessati.
La chiave di ripartizione costi può essere per esempio la dimensione del Cantone interessato o il numero di utenti.
e) **Contributo**: se l'autorità federale presta solo un contributo di personale a un progetto esistente.

In linea di massima la variante d (chiave di ripartizione costi) ha senso solo se è presente anche una gestione congiunta del prodotto come nella variante b del coinvolgimento dei fornitori. Soltanto chi ha diritto di codecisione sarà anche disposto ad assumersi i costi conseguenti alle decisioni.

5.4 Supporto

Nel contesto della ripartizione dei costi e del dispendio è importante chiarire chi fornisce il supporto. L'articolo 9 LMeCA ammette un rilascio minimo. In alcune circostanze, questa variante potrebbe non sortire l'effetto auspicato e non costituisce di per sé una community in cui l'autorità federale ha influenza.

a) **Rilascio minimo**: nessun supporto (ovvero in generale il rilascio avviene senza un supporto organizzato e senza la partecipazione dell'autorità federale).
b) **Supporto della Confederazione**: l'autorità federale (o il suo fornitore di prestazioni) mette a disposizione il supporto definito.
c) **Supporto di terzi**: l'autorità o la community definiscono un'organizzazione di supporto.

L'impatto che si vuole ottenere, per esempio sull'ecosistema Svizzera o sul finanziamento iniziale di una community, può far sì che sia la Confederazione a fornire il supporto. Ovviamente occorre definire anche il livello di supporto e in linea di massima è possibile richiedere una rimunerazione. A causa dei meccanismi della Confederazione, i fornitori o i fornitori di prestazioni sono maggiormente in grado di offrire supporto commerciale.

Secondo la LMeCA, gli uffici possono offrire supporto dietro pagamento (eventualmente anche tramite fornitori di prestazioni e terzi). A tale scopo, poiché sussiste la base legale necessaria, devono pubblicare i rispettivi emolumenti e tariffe sulla loro pagina web. Il collegamento al sistema di pagamento può essere discusso con il DFF.

Esempi di supporto ed emolumenti sono indicati nell'appendice G.

5.5 Sviluppo

Il metodo di base dello sviluppo può influire anche sul tipo di community.

a) **Interno**: l'utilizzo di «repository» interni comporta che non vi possano collaborare direttamente terzi. Più è rigida la separazione, tanto maggiore sarà lo sforzo di

integrazione da parte della Confederazione laddove voglia inserire da sé ampliamenti di terzi. Anche il rilascio stesso è più dispendioso.

b) **Terzi**: il fornitore lavora per conto proprio oppure il progetto già avviato o gestito da terzi viene sviluppato secondo regole definite.
c) **Aperto**: lo sviluppo di software avviene direttamente in un «repository» pubblico. Secondo questa modalità, il rilascio secondo LMeCA è garantito. I costi devono essere sostenuti durante lo sviluppo. In questo scenario è possibile inoltre collaborare prima (e in modo più semplice) con terzi.

6 Contenuti dettagliati del piano per la community

Il presente numero offre un ausilio alla stesura del piano per la community in base alle decisioni di principio prese (secondo il n. 5).

In base alle decisioni di principio prese sono opportuni contenuti diversi.

Gestione del prodotto	Il fornitore è	Ripartizione costi	Proposta
Aperta	Partner	Confedera-zione	Contenuti o spunti per decisioni di principio che corrispondono a: Gestione del prodotto: c) organizzazione aperta Coinvolgimento del fornitore: b) il fornitore è partner Ripartizione dei costi: b) ognuno per sé *Il testo di cui sopra può essere copiato nel modello e adattato.*
Congiunta	-	Ognuno per sé	Contenuti o spunti per decisioni di principio che corrispondono a: Gestione del prodotto: b) elaborazione congiunta Coinvolgimento del fornitore: a) oppure b) (non rilevante) Ripartizione dei costi: c) principio di causalità oppure d) chiave di ripartizione costi
Congiunta	-	Respon-sabile	

Il piano per la community deve quindi essere un documento dinamico che viene adattato d'intesa con la community stessa. Il piano deve rappresentare la situazione attuale e proporre opzioni di sviluppo futuro. Nel caso in cui successivamente accedano alla community più membri, è possibile elaborare ulteriormente l'organizzazione e i processi e introdurre meccanismi più complessi. I numeri seguenti si orientano direttamente alla struttura proposta per il piano per la community (v. n. 4).

Se la gestione del prodotto è in mano a terzi, saranno loro a definire i piani per la community.

6.1 Informazioni principali

La seguente lista di informazioni principali andrebbe redatta e pubblicata per ogni community, così da permettere a potenziali interessati al software e alla community di presentarsi:

- Nome del software
- Repository
- Titolare
- Unità organizzativa / ufficio responsabile
- Indirizzo di contatto
- Licenza utilizzata
- Livello di supporto
- Progetto esistente

Queste informazioni possono essere rappresentate secondo lo standard publicode.yaml[13].

6.2 Finalità

Da un lato, la finalità deve spiegare qual è lo scopo effettivo o la «visione» dell'applicazione. Tale visione deve confluire anche nel file **README.md** nel «repository» (cfr. «Em002-2.2 Lista di controllo Analisi e preparazione»).

Dall'altro lato, si deve descrivere cosa si persegue dal punto di vista della community:

- Chi deve accedervi? Chi sono i possibili interessati?
- Occorre cercare attivamente nuovi membri?
- Quali obiettivi principali si perseguono con la pubblicazione come open source?

6.3 Organizzazione e governance

In linea di massima si mira a far sì che, per i nuovi sviluppi, la Confederazione sia titolare dei diritti di base sul codice sorgente.

Il tipo di gestione del progetto dipende da quale organizzazione è responsabile (SAFe, HERMES) nel caso della Confederazione. Per quanto riguarda la forma organizzativa si dovrebbe sempre prestare attenzione a un «overhead» minimo (l'esistente ha priorità rispetto al nuovo, la società semplice rispetto all'associazione).

Gestione del prodotto	Il fornitore è	Sviluppo	Proposta
Confedera-zione	-	Interno	Essendo la Confederazione stessa a mantenere direttamente il controllo, non è necessario costituire organizzazioni aggiuntive. Il servizio che si occupa della pubblicazione (p. es. un ufficio) mantiene la proprietà e si assume tutti i compiti di coordinamento. In questo caso, il capitolo sull'organizzazione può essere conciso.

[13] https://yml.publiccode.tools/

Congiunta	-	Interno	Per questa casistica è consigliabile perlopiù un'organizzazione come **società semplice**. Affinché la roadmap e i requisiti possano essere elaborati e concordati insieme, si raccomanda di creare due gruppi in cui sia presente un rappresentante ciascuno per ogni membro della community: • Consiglio direttivo: definisce la strategia e la definizione delle priorità. • Gruppo specialistico: elabora i requisiti e i contenuti concreti a livello di esperti. A seconda delle dimensioni, si può anche impiegare un gruppo specialistico separato per ogni argomento. Il proprietario del codice è l'ufficio che l'ha pubblicato. Se le strutture si fanno più complesse o se sono coinvolti più di cinque utenti o altri membri si può verificare se non sia meglio organizzare la community come associazione. A tal fine si possono confrontare i dettagli della variante esposta qui di seguito.
Congiunta	-	Terzi	Si raccomanda un'organizzazione sotto forma di **associazione**. O si crea una propria associazione appositamente per l'applicazione oppure si cede l'applicazione a un'associazione esistente. Si ricorre a un'associazione perché diversamente la ripartizione dei costi diventa dispendiosa e probabilmente nell'ambito dei compiti di coordinamento si genera abbastanza lavoro per impiegare a tempo parziale un amministratore per l'associazione. All'associazione vengono poi ceduti i diritti sul codice, l'amministratore assume la gestione della community e si occupa delle ordinazioni presso il fornitore.
Congiunta	Mandatario	Interno	Si raccomanda un'organizzazione sotto forma di **associazione**. O si crea una propria associazione appositamente per l'applicazione oppure si cede l'applicazione a un'associazione esistente. Si ricorre a un'associazione perché diversamente la ripartizione dei costi diventa dispendiosa e probabilmente nell'ambito dei compiti di coordinamento si genera abbastanza lavoro per impiegare a tempo parziale un amministratore per l'associazione. All'associazione vengono poi ceduti i diritti sul codice, l'amministratore assume la gestione della community e si occupa delle ordinazioni presso il fornitore. In questo caso i fornitori non sono membri dell'associazione ma solo mandatari. L'amministratore non deve essere designato da un fornitore.

Congiunta	Mandatario	Terzi	Si raccomanda un'organizzazione sotto forma di **associazione**. O si crea una propria associazione appositamente per l'applicazione oppure si cede l'applicazione a un'associazione esistente. Si ricorre a un'associazione perché diversamente la ripartizione dei costi diventa dispendiosa e probabilmente nell'ambito dei compiti di coordinamento si genera abbastanza lavoro per impiegare a tempo parziale un amministratore per l'associazione. All'associazione vengono poi ceduti i diritti sul codice, l'amministratore assume la gestione della community e si occupa delle ordinazioni presso il fornitore. I fornitori sono membri dell'associazione e l'amministratore può essere designato anche da un fornitore.
Aperta	-	Interno	**Non** serve una **lunga descrizione di un'organizzazione** che viene concepita deliberatamente come aperta. Tuttavia occorre disciplinare chi riceve le richieste provenienti dall'esterno e le elabora (responsabilità). Si deve inoltre definire se e in che misura possano essere coinvolte persone esterne: • Nessun coinvolgimento diretto: gli esterni possono apportare solo proposte e contributi (sotto forma di «pull request»). • Coinvolgimento come «contributor»: gli esterni che prestano contributi buoni e frequenti vengono coinvolti direttamente. • Possibilità di trasferimento a esterni: se, dopo la pubblicazione, gli esterni contribuiscono all'ulteriore sviluppo in forma maggiore rispetto all'ufficio corrispondente, è possibile cedere loro l'applicazione. L'ufficio che si occupa della pubblicazione mantiene la proprietà. Questa variante corrisponde all'organizzazione «classica» dei progetti open source; per ulteriori informazioni si veda per esempio https://opensource.guide/leadership-and-governance/.

Aperta	Mandatario	Interno	**Non** serve una **lunga descrizione di un'organizzazione** che viene concepita deliberatamente come aperta. Tuttavia occorre disciplinare chi riceve le richieste provenienti dall'esterno e le elabora (responsabilità). Si deve inoltre definire se e in che misura possano essere coinvolte persone esterne: • Nessun coinvolgimento diretto: gli esterni possono apportare solo proposte e contributi (sotto forma di «pull request»). • Coinvolgimento come «contributor»: gli esterni che prestano contributi buoni e frequenti vengono coinvolti direttamente. • Possibilità di trasferimento a esterni: se, dopo la pubblicazione, gli esterni contribuiscono all'ulteriore sviluppo in forma maggiore rispetto all'ufficio corrispondente, è possibile cedere loro l'applicazione. L'ufficio che si occupa della pubblicazione mantiene la proprietà. Questa variante corrisponde all'organizzazione «classica» dei progetti open source; per ulteriori informazioni si veda per esempio https://opensource.guide/leadership-and-governance/. Il fornitore dovrebbe assumersi solo compiti di coordinamento di natura tecnica (revisione del codice, valutazione).
Terzi	Partner	Interno	**Non** serve una **lunga descrizione di un'organizzazione** che viene concepita deliberatamente come aperta. Tuttavia occorre disciplinare chi riceve le richieste provenienti dall'esterno e le elabora (responsabilità). Si deve inoltre definire se e in che misura possano essere coinvolte persone esterne: • Nessun coinvolgimento diretto: gli esterni possono apportare solo proposte e contributi (sotto forma di «pull request»). • Coinvolgimento come «contributor»: gli esterni che prestano contributi buoni e frequenti vengono coinvolti direttamente. • Possibilità di trasferimento a esterni: se, dopo la pubblicazione, gli esterni contribuiscono all'ulteriore sviluppo in forma maggiore rispetto all'ufficio corrispondente, è possibile cedere loro l'applicazione. L'ufficio che si occupa della pubblicazione mantiene la proprietà. Questa variante corrisponde all'organizzazione «classica» dei progetti open source; per ulteriori informazioni si veda per esempio https://opensource.guide/leadership-and-governance/. Il fornitore può assumere compiti di coordinamento.

Sulla governance si può consultare anche [BITKOM2023] n. 4.1, [IzCab2023] oppure https://github.com/todogroup/ospo-career-path/blob/main/OSPO-101/module7/README.md#governance-models.

Statuti di esempio sono riportati nell'appendice E.

Attenzione: dato che deve essere garantito il rilascio ai sensi dell'art. 9 LMeCA, l'Amministrazione federale deve assicurarsi che la pubblicazione non possa essere ritirata senza preavviso (cfr. capoverso V.2 del documento «Guida per l'acquisto di OSS» [BBL-WL]).

6.4 Roadmap e processo di modifica

Gestione del prodotto	Il fornitore è	Sviluppo	Proposta
Confe-derazion	Mandatario	-	L'autorità federale definisce la roadmap e decide riguardo a quali modifiche implementare. A tal fine si applicano i processi standard dell'ufficio interessato.
Confe-derazion	Partner	-	L'autorità federale definisce la roadmap coinvolgendo l'azienda interessata. L'autorità federale decide quali modifiche implementare. A tal fine si applicano i processi standard dell'ufficio interessato.
Congiunta	-	Interno	Il processo di elaborazione della roadmap e di approvazione delle modifiche deve essere deciso dai membri della società o dell'associazione. A seconda della struttura dei membri (numero, proporzione ecc.) possono essere opportuni altri processi. I processi devono però contenere misure per la risoluzione dei conflitti e l'ottenimento della maggioranza.
Congiunta	-	Terzi	Il processo di elaborazione della roadmap e di approvazione delle modifiche deve essere deciso dai membri della società o dell'associazione. A seconda della struttura dei membri (numero, proporzione ecc.) possono essere opportuni altri processi. I processi devono però contenere misure per la risoluzione dei conflitti e l'ottenimento della maggioranza. A questo si aggiunge che si deve stabilire anche una chiave di ripartizione dei costi. Occorre anche tenere in considerazione come si debba gestire la ripartizione dei costi per modifiche non auspicate dai membri interessati. Soprattutto in associazioni formate da membri di dimensioni diverse possono verificarsi situazioni in cui un membro più grande (p. es. l'autorità federale) si assume una parte consistente dei costi per un ampliamento senza trarne però alcun vantaggio.
Aperta	-	-	Esempio, non adatto a tutte le situazioni. Si rinuncia alla stesura di una roadmap, l'applicazione soddisfa al momento le esigenze. Le modifiche vengono decise in una discussione aperta in cui si cerca il consenso. Il voto decisivo spetta al proprietario del codice (Confederazione Svizzera).

6.5 Processo di sviluppo

L'eventuale ricorso all'«open development» va definito qui. La distribuzione dei compiti e l'indicazione dei nomi dei servizi centrali nel piano per la community sono importanti, così come il modo in cui viene garantita la sicurezza dei contenuti del «repository».

Gestione del prodotto	Il fornitore è	Sviluppo	Proposta
Confederazione	-	-	I diritti di «committer» (diritti di scrittura sul codice) spettano alle persone/servizi/fornitori scelti dalla Confederazione. Dopo lo scadere del contratto, tali diritti vengono di nuovo revocati. I fornitori e i fornitori di prestazioni sono responsabili di eseguire il processo di revisione del codice tecnico per i contributi esterni che la Confederazione ha accettato dal punto di visto specialistico. Essi sono responsabili della qualità tecnica. Per quest'attività, i fornitori vengono compensati dalla Confederazione.
Congiunta	Mandatario	Interno	In linea di massima tutti i servizi incaricati da un membro della community ricevono diritti di scrittura sul codice (diritti di «committer»). Dopo lo scadere del contratto, tali diritti vengono di nuovo revocati. Se il codice modificato riguarda parti essenziali dell'applicazione, tutte le modifiche devono essere revisionate da un servizio specificamente incaricato dalla community. Tale servizio è responsabile della qualità tecnica dell'applicazione e ha anche il compito di revisionare le modifiche che la community ha accettato dal punto di vista specialistico.
Congiunta	Partner	Terzi	I diritti di «committer» (diritti di scrittura sul codice) spettano ai membri della community. Queste persone si organizzano in autonomia e si assumono insieme la responsabilità per la qualità del codice. Se un membro della community incarica un fornitore esterno di eseguire una modifica o un ampliamento, allora seguirà una revisione di tale modifica da parte di un membro della community. Questi riceverà un compenso dal committente. Le modifiche apportate al nucleo dell'applicazione devono essere revisionate da un secondo membro della community. I fornitori che sono membri della community sono a loro volta responsabili della revisione dei contributi esterni e del nuovo codice tecnico che l'associazione ha accettato dal punto di vista specialistico.

Aperta	-	-	I diritti di «committer» spettano inizialmente agli sviluppatori o ai loro datori di lavoro. Vengono conferiti diritti di «committer» a tutti gli sviluppatori che durante diversi mesi contribuiscono attivamente al progetto e dimostrano un'adeguata competenza sociale. I singoli sviluppatori mantengono in genere i propri diritti di «committer» anche se cambiano datore di lavoro o la Confederazione sceglie un altro fornitore. I diritti di «committer» decadono solo se vengono ceduti volontariamente oppure se la maggioranza degli altri «committer» vota per la revoca. La Confederazione ha il diritto di nominare «committer» singoli sviluppatori delle aziende da essa incaricate. Non ha invece il diritto di revocare a qualcuno i diritti di «committer» senza una motivazione. Le revisioni del codice vengono svolte da almeno una persona che svolge il ruolo di «committer» e i «committer» si organizzano in autonomia. La Confederazione riconosce di compensare il lavoro di revisione purché sia per essa finanziabile.

6.6 Ripartizione dei costi e fornitori

Gestione del prodotto	Il fornitore è	Sviluppo	Proposta
Confederazione	Mandatario	Interno	I fornitori vengono scelti periodicamente tramite bando OMC o procedura per incarico diretto (in base all'entità delle prestazioni di manutenzione).
Confederazione	Partner	-	Importante: verificare preventivamente se nel caso concreto questo scenario sia ammissibile sotto il profilo del diritto in materia di appalti pubblici. Su questo punto seguiranno ulteriori integrazioni nella documentazione. La Confederazione sceglie il fornitore in questione come partner strategico e mira a una collaborazione a lungo termine.

Congiunta	Mandatario	Interno	Ogni membro della community incarica per conto proprio uno o più fornitori di software per l'attuazione delle modifiche e degli ampliamenti di suo interesse. Per modifiche auspicate da più membri si concorda una ripartizione dei costi caso per caso. Il membro che si assume la quota maggiore dei costi è responsabile della selezione e dell'assegnazione dell'incarico al fornitore. (Regolamentazione sulla manutenzione del software)
Congiunta	Mandatario	Terzi	L'associazione sceglie a livello centralizzato i fornitori di software periodicamente tramite bando OMC o procedura per incarico diretto (in base all'entità delle prestazioni di manutenzione). I costi vengono ripartiti fra i membri in base a un'apposita chiave (da definire: in base alla popolazione, agli utenti ecc.).
Congiunta	Partner	Interno	Ogni membro della community incarica per conto proprio uno o più fornitori di software per l'attuazione delle modifiche e degli ampliamenti di suo interesse. Per modifiche auspicate da più membri si concorda una ripartizione dei costi caso per caso. Il membro che si assume la quota maggiore dei costi è responsabile della selezione e dell'assegnazione dell'incarico al fornitore. *(Regolamentazione sulla manutenzione del software)* Eventualmente con integrazione di un contributo dei fornitori partecipanti. Esempio: *Ogni azienda di sviluppo software membro della community si impegna a investire annualmente a proprie spese almeno XY giorni di lavoro nell'ulteriore sviluppo, nell'aggiornamento tecnologico o nella commercializzazione dell'applicazione.*
Congiunta	Partner	Terzi	L'associazione sceglie a livello centralizzato i fornitori di software periodicamente tramite bando OMC o procedura per incarico diretto (in base all'entità delle prestazioni di manutenzione). I costi vengono ripartiti fra i membri in base a un'apposita chiave (da definire: in base alla popolazione, agli utenti ecc.). Eventualmente con integrazione dell'aggiunta: *Ogni azienda di sviluppo software membro della community si impegna a investire annualmente a proprie spese almeno XY giorni di lavoro nell'ulteriore sviluppo, nell'aggiornamento tecnologico o nella commercializzazione dell'applicazione.*

Terzi	Mandatario	-	I fornitori vengono scelti periodicamente tramite bando OMC o procedura per incarico diretto (in base all'entità delle prestazioni di manutenzione). Con l'aggiunta: *I miglioramenti apportati dai potenziali fornitori e le attività nella community vengono presi in considerazione come fattori rilevanti nella valutazione.* Le modifiche auspicate da esterni non vengono finanziate dalla Confederazione, bensì devono essere prese a carico e corrisposte da essi stessi.
Terzi	Partner	-	Ogni membro della community incarica per conto proprio uno o più fornitori di software per l'attuazione delle modifiche e degli ampliamenti di suo interesse. Per modifiche auspicate da più membri si concorda una ripartizione dei costi caso per caso. Il membro che si assume la quota maggiore dei costi è responsabile della selezione e dell'assegnazione dell'incarico al fornitore. (Regolamentazione sulla manutenzione del software) Con l'aggiunta: *Dal fornitore ci si aspetta una collaborazione attiva nella community anche se il dispendio che ne deriva non viene direttamente compensato.* *Le modifiche auspicate da esterni non vengono finanziate dalla Confederazione, bensì devono essere prese a carico e corrisposte da essi stessi. Il fornitore si dichiara disposto ad attuarle su richiesta dell'ente esterno a prezzi onesti.*

6.7 Commercializzazione

Gestione del prodotto	Il fornitore è	Sviluppo	Proposta
Confederazione	Mandatario	-	La Confederazione non intraprende attività di commercializzazione dell'applicazione, che viene però pubblicata sulle piattaforme per software a codice sorgente aperto. Negli organi specializzati si richiama l'attenzione sulla pubblicazione dell'applicazione. Se si presentano persone interessate, si verifica se sia opportuno fondare una community congiunta o se continuare a lavorare con la versione del software esistente.
Congiunta	Partner	-	Si integra a quanto sopra. I fornitori possono commercializzare l'applicazione e cercare ulteriori interessati. La Confederazione può essere indicata come referenza. Accettiamo consapevolmente che in questo caso potrebbero essere create versioni divergenti del software.
Con-giunta	Mandatario	Interno	L'applicazione viene commercializzata dai membri o dall'associazione.
Con-giunta	Partner	Interno	L'applicazione viene commercializzata dai fornitori.
Con-giunta	-	Terzi	L'applicazione viene commercializzata dall'associazione, dalla società semplice e dai relativi membri.
Aperta	-	-	L'applicazione viene commercializzata dai fornitori oppure si rinuncia a una commercializzazione esplicita. Variante senza commercializzazione esplicita: La Confederazione non intraprende attività di commercializzazione dell'applicazione, che viene però pubblicata sulle piattaforme per software a codice sorgente aperto. Negli organismi specializzati richiamiamo l'attenzione sul fatto che l'applicazione è stata pubblicata e che è gradita la collaborazione. Presentiamo il software agli interessati e cerchiamo di coinvolgerli nei nostri processi di concezione della roadmap.

6.8 Gestione dei contributi esterni

Gestione del prodotto	Il fornitore è	Sviluppo	Proposta
Confederazione	-	-	I contributi e le richieste esterne devono essere elaborati anche se la Confederazione resta l'unico decisore sul software (tranne nel caso del rilascio minimo, in cui è però inevitabile ricorrere a un «fork»). Proponiamo il seguente contenuto: *Gestione degli errori segnalati* *Cerchiamo di riprodurre gli errori segnalati da esterni e, se necessario, chiediamo ulteriori precisazioni. Se la riproduzione riesce, la Confederazione commissiona l'eliminazione dell'errore. Se non riusciamo a capire l'errore o il dispendio per eliminarlo è eccessivamente elevato, ci scusiamo con chi ha segnalato l'errore e chiediamo se gli sia possibile effettuare una «pull request» per l'eliminazione.* *Gestione delle richieste* *Reagiamo a ogni richiesta in entrata. Se non è possibile chiarire la richiesta con uno sforzo ragionevole e non risulta efficiente impiegarvi altro tempo, ci scusiamo indicando che i nostri mezzi sono limitati e proponiamo al richiedente di mettersi in contatto con un fornitore per un'ulteriore consulenza.* *Gestione delle «pull request»* *Verifichiamo di buon grado ogni «pull request». Onde evitare un dispendio inutile da parte dei «contributor» segnaliamo fin dall'inizio se qualcosa va in contrasto con la nostra roadmap o se non siamo certi che accetteremo il contributo. La Confederazione decide in autonomia e in base a criteri tecnici quali contribuiti accettare. Si accettano sempre le correzioni di errori che hanno superato il processo di revisione.* *Gestione dei «fork»* *Supportiamo i «fork» della nostra applicazione. Anche chi impiega l'applicazione deve a sua volta presupporre un «fork». Osserviamo almeno una volta all'anno i «fork» che sono stati creati e riprendiamo eventuali ampliamenti di buona qualità e adatti a noi.*

Congiunta	-	-	«Esterno» si riferisce qui ai contributi provenienti dal di fuori della community definita (ossia i membri interni all'associazione). Divergenze rispetto a quanto sopra: *Gestione dei «fork»* *Miriamo a evitare che si creino dei «fork» (di lunga durata) dell'applicazione. Al contrario, le persone interessate devono accedere direttamente alla community. All'interno della community cerchiamo di far sì che tutti i membri utilizzino la versione principale.*
Aperta	-	-	*Per noi è importante la creazione di una community funzionante e di una cultura aperta.* *Gestione degli errori segnalati* *Cerchiamo di riprodurre e correggere gli errori segnalati. A tal fine richiediamo la collaborazione attiva delle persone che effettuano la segnalazione. Se possibile per loro, devono redigere direttamente una «pull request» con correzione dell'errore.* *Gestione delle richieste* *Elaboriamo ogni richiesta entro al massimo una settimana. Elaboriamo in via prioritaria e approfondiamo le richieste dei «contributor» attivi. Cerchiamo di coinvolgere nell'elaborazione della richiesta altri partecipanti.* *Gestione delle «pull request»* *Per quanto possibile, cerchiamo di ricevere molte «pull request» e valide. Nelle revisioni e nella valutazione tecnica e specialistica delle «pull request» coinvolgiamo tutti i «contributor» attivi. In caso di contributi controversi cerchiamo di prendere una decisione con una votazione informale («voting»).* *Gestione dei «fork»* *Per quanto possibile, cerchiamo di evitare che si rendano necessari dei «fork», in quanto gestiamo la community in modo attivo e aperto. Se si generano «fork», li consideriamo in maniera attiva e cerchiamo di restituire gli ampliamenti e i miglioramenti che ne derivano. In questo caso, richiediamo attivamente «pull request».*

In generale, si deve considerare che i contributi possono variare dal supporto ad altri utenti fino alla responsabilità per intere parti di un progetto (cfr. [BITKOM2023] n. 4.2.

6.9 Funzionamento dell’applicazione

Gestione del prodotto	Il fornitore è	Sviluppo	Proposta
Congiunta	-	-	In via opzionale, il funzionamento può essere offerto aggiuntivamente a livello centralizzato dall’associazione centrale nell’ottica di «software as a service». Si deve verificare se ciò corrisponde a quanto auspicato.
-	Partner	-	Stabilire se il fornitore deve eseguire il software immediatamente.
-	Direzione	-	Stabilire che il fornitore è responsabile del funzionamento.

In questo numero si devono stabilire anche i livelli e i contatti di supporto.

7 Note importanti per i responsabili di una community

7.1 Processo a lungo termine

Costruire una community di successo è un processo a lungo termine che richiede prerseveranza e costanza.

7.2 Ulteriori spunti per la creazione di community

Ulteriori spunti per una buona gestione della community sono disponibili agli indirizzi https://opensource.guide/code-of-conduct/ e https://opensource.guide/best-practices/.

7.3 Comunicazione

Con la pubblicazione di software e documentazione si apre un ulteriore canale di comunicazione dell'Amministrazione federale. Se ciò non avviene in maniera accurata e professionale, la pubblica reputazione della Confederazione ne risente.

Dal punto di vista della comunicazione, nell'Amministrazione federale le community OSS si trovano in una situazione di conflitto. Da un lato, si applicano le **direttive generali per la comunicazione**, dall'altro la community deve anche cercare una semplice **collaborazione formale e informale** al di là dei confini dell'organizzazione, per quanto possibile senza controlli di qualità e rilasci formali.

L'utilizzo di «repository» pubblici e, in determinate circostanze, di tracciamento delle problematiche («issue tracking») e wiki pubblici fa sì che le dichiarazioni dei singoli collaboratori al progetto siano visibili al pubblico senza filtri. Questo comporta che, in un contesto simile, i collaboratori al progetto siano invitati a comportarsi in maniera professionale e che si debbano rispettare gli standard di qualità per le pubblicazioni.

In progetti OSS, questo viene solitamente disciplinato dal «**code of conduct**».
Lo stesso codice di comportamento dell'Amministrazione federale[14] è formulato in termini generali; tuttavia, in sostanza, si può applicare anche qui il principio:
«Staff carry out their duties with a sense of responsibility, integrity and loyalty. In their private lives they also take care not to tarnish the reputation, the credibility and the image of the Confederation» / «Nello svolgimento della loro attività professionale, i dipendenti si comportano in maniera responsabile, integra e leale. Si assicurano di difendere la buona reputazione, l'immagine e la credibilità dell'Amministrazione federale anche nella vita privata».

7.4 Sviluppo aperto

Se lo sviluppo di un software è programmato sin dal principio in un «repository» aperto, il rilascio avviene in maniera automatica. A tal fine le liste di controllo corrispondenti secondo le istruzioni [Em002-2] devono essere compilate già all'inizio e i processi di sviluppo del fornitore (di prestazioni) devono renderlo possibile. Il vantaggio è che si può creare la community sin da subito. Possono collaborare allo sviluppo anche più organizzazioni.

Un repository chiuso è invece una forma mista che viene utilizzata da altri partner nell'ambito di una collaborazione.

[14] www.epa.admin.ch/dam/epa/fr/dokumente/aktuell/medienservice/120_verhaltenskodex_e.pdf.download.pdf/120_verhaltenskodex_e.pdf

7.5 Gestione utenti

È il progetto (o l'organizzazione che fornisce il supporto) a stabilire se gli sviluppatori e gli altri collaboratori al progetto lavorano utilizzando il proprio nome o in forma anonima. Se le persone non lavorano in forma anonima e dispongono già di un login alle piattaforme, è più semplice usare tali piattaforme (per scopi professionali o privati).

In linea di principio la piattaforma dovrebbe permettere la registrazione di terzi, così da rendere possibile la collaborazione anche da parte di sviluppatori esterni alla Confederazione.

Al momento dell'uscita dal progetto, i rispettivi diritti devono essere revocati. Ne è responsabile la gestione del prodotto.

7.6 Raggruppamento delle community

Se possibile, il numero di community deve essere mantenuto ridotto. Ciò significa che:

- se esiste già una community in cui è prevista una struttura simile per problemi analoghi e con le stesse persone, allora si dovrebbe utilizzare una sola community;
- i progetti simili e che si rivolgono agli stessi destinatari possono essere raggruppati in un'unica community;
- in linea di principio, bisognerebbe creare nuove strutture di community solo se strettamente necessario.

7.7 Gestione dei contributi di terzi

Il tema dei contributi di software ad altri progetti è trattato nel capitolo 3.3. Queste disposizioni si applicano analogamente anche nel caso di natura opposta.

Punti importanti:

- Un CLA adeguato garantisce che i diritti del codice sorgente spettino alla Confederazione (cfr. anche la sezione «Contributor License Agreeement e Developer Certificate of Origin» del documento «Em002-3 Guida Licenze OSS»). Il CLA deve essere firmato dal titolare del contributo.
- Le conferme vanno archiviate.
- Il contributo va verificato prima di integrarlo nel codice di base.
- Bisognerebbe rispondere tempestivamente e cortesemente alle richieste (problemi, domande, richieste di pull).
- I canali e la governance vanno indicati nel piano per la community, sul sito web e nel repository.

7.8 Segnalazioni rilevanti per la sicurezza

Oltre alla normale segnalazione di errori, se il tipo di progetto lo richiede, dovrebbe essere data la possibilità di segnalare in modo affidabile gli errori rilevanti per la sicurezza, come previsto per esempio in un programma «bug bounty».

Può fungere da esempio la documentazione di trustbroker.swiss:

- https://github.com/trustbroker-swiss/trustbroker.swiss/blob/main/Security-and-Vulnerability-Disclosure.md

7.9 Limitazione dei «fork»

Nel mondo dell'open source, i «fork» sono una conseguenza piuttosto naturale. Ciononostante, la reintegrazione del codice sorgente e delle caratteristiche da un «fork» non è un'operazione banale. Eventualmente è opportuno cercare un dialogo con le persone o istituzioni che vogliono prendere in considerazione un «fork» e cercare di persuaderle a rimanere sul progetto principale. In questi casi è spesso necessario trovare un compromesso.

7.10 Ripartizione dei costi

Può essere opportuno concordare chiavi di ripartizione generali tra i partner più importanti. Queste possono essere adattate ogni paio d'anni se la situazione cambia radicalmente. Affinché il lavoro sul software si svolga in maniera efficiente, i partner maggiori dovrebbero tendenzialmente assumersi una percentuale leggermente maggiore a quella che gli spetterebbe (ed eventualmente anche il coordinamento generale). L'organizzazione deve comunque lavorare e non discutere sulle questioni finanziarie. Se la collaborazione comprende più progetti (si possono addurre come esempio le soluzioni settoriali nel campo delle ferrovie a scartamento normale) si dovrebbe utilizzare una chiave di ripartizione standardizzata per tutti i progetti. Dal punto di vista delle autorità federali è già un buon miglioramento non doversi assumere da sole i costi.

Possibili criteri per le chiavi di ripartizione:

- valutazione della distribuzione dell'utile (p. es. tra l'autorità federale e un fornitore che spera in altri clienti);
- numero di abitanti / utenti (p. es. tra uffici, Comuni, Città);
- forza economica (p. es. Cantoni);
- prodotto sociale lordo (p. es. tra Cantoni);
- chi vuole avere più controllo

Onde evitare discussioni, la chiave di ripartizione non dovrebbe dipendere semplicemente dal budget annuale. In determinate circostanze, una clausola secondo il principio di causalità può consentire adattamenti più rapidi per alcuni partner, se sono disposti a pagare extra.

La chiave di ripartizione dovrebbe essere applicata anche per l'«overhead», la manutenzione / il supporto e un minimo dello sviluppo ulteriore. Se possibile, questi elementi dovrebbero essere risolti sul lungo termine.

È bene notare che i terzi che contribuiscono finanziariamente esigono anche una maggiore influenza. Il community management deve essere in grado di gestire una situazione di questo tipo.

7.11 Prestazioni di servizi complementari secondo l'articolo 9 capoversi 5 e 6 LMeCA

Le autorità federali possono (esse stesse, tramite fornitori o fornitori di prestazioni) fornire prestazioni di servizio complementari, in particolare all'integrazione, alla manutenzione, alla garanzia della sicurezza delle informazioni e al supporto.

Nel farlo, devono rispettare la restrizione per cui tali prestazioni servano all'adempimento dei compiti delle autorità e possano essere fornite con un onere proporzionato.

Ne consegue che l'autorità federale non può essere l'azienda sviluppatrice per terzi. Essa pone rimedio agli errori, può occuparsi di aspetti rilevanti per la sicurezza, può fornire una

parte del supporto e aiuto nell'integrazione (tutte attività che contribuiscono alla creazione di una community). I nuovi sviluppi e le caratteristiche aggiuntive per terzi dovrebbero però concentrarsi innanzitutto sul normale scopo dell'attività dell'autorità. Ovviamente può anche darsi che l'approntamento del software faccia parte dei suoi compiti e in tal caso ci si può dedicare in qualsiasi momento agli sviluppi del software. È anche vero, tuttavia, che il compito principale dell'autorità non è lo sviluppo di software.

Ciò significa che se si sviluppano caratteristiche sostanziali esclusivamente per terzi, queste devono essere integrate in collaborazione con i partner oppure dai fornitori.

Affinché questo non causi problemi a livello di diritto in materia di appalti pubblici, nell'acquisto di software e prestazioni di servizi si deve prestare attenzione al fatto che i fornitori possano fornire le caratteristiche anche a terzi (in alcune circostanze solo terzi statali).

La definizione di controprestazioni è stata già definita nella sezione relativa al supporto. La LMeCA costituisce già la base legale per tali emolumenti, che devono essere comunicati sul sito web dell'autorità federale in questione. L'appendice G riporta alcuni esempi.

L'articolo 9 capoverso 6 statuisce che, nel caso normale, non si deve entrare in concorrenza con l'economia privata. Ciò significa che gli emolumenti non possono essere fissati troppo al ribasso e devono almeno coprire i costi. In caso di dubbio, piuttosto è opportuno fissare le tariffe orarie verso l'alto.

In generale, si consiglia di lavorare per quanto possibile con 1–3 tariffe orarie diverse.

Se un dipartimento vuole definire deroghe al capoverso 6, non si deve entrare in concorrenza con l'economia privata. In certe circostanze, il modo più semplice è fissare gli emolumenti ed eventualmente aumentarli in caso di obiezione da parte di aziende dell'economia privata che sono effettivamente in grado di offrire la prestazione.

7.12 Sovranità digitale

Fintanto che la Confederazione non dispone di un proprio «repository», per garantire la sovranità digitale si consiglia di creare regolarmente copie del «repository» pubblico.
Nel farlo si deve fare attenzione a non copiare solo il codice sorgente, ma anche le ulteriori informazioni come «issue tracking», wiki, configurazioni ecc.
Inoltre si deve risolvere la gestione utenti per garantire il controllo sul codice sorgente pubblicato. L'obiettivo sovraordinato è garantire uno sviluppo indipendentemente dalla piattaforma in questione e dare alla community la possibilità di cambiare[15].

[15] Cfr. anche www.bfh.ch/de/aktuell/news/2024/neue-studie-digitale-souveraenitaet/

Appendice

A. Modifiche rispetto alla versione precedente

- Nuovo capitolo 1 «L'essenziale in breve»
- Capitolo 3: aggiunta di informazioni relative allo scopo di una community
- Nuovi capitoli 3.1, 3.2 e 3.3 dedicati al tema della collaborazione
- Nuovo capitolo 7.7 «Gestione dei contributi di terzi»
- Reso attenti che i fornitori non dovrebbero pubblicare da soli (secondo [BBL-WL], V.2).
- Aggiunta di informazioni relative a publicode.yaml
- Confronto con il nuovo documento [Em002-7]
- Piccole modifiche redazionali e miglioramenti

B. Riferimenti

Cfr. «Em002 Guida strategica Software a codice sorgente aperto nell'Amministrazione federale».

C. Abbreviazioni

Cfr. «Em002 Guida strategica Software a codice sorgente aperto nell'Amministrazione federale» e «Em002-6 Domande frequenti sulla pubblicazione di OSS (FAQ OSS)».

D. Esempi di piani per community esistenti

Nell'ottica di raccogliere buone pratiche, di seguito sono elencati alcuni piani per community già esistenti e documenti paragonabili. Si tratta non necessariamente di piani di autorità federali, bensì anche di altre organizzazioni.

- Community GERES (http://geres-community.ch)
  Classificazione:
  - Gestione del prodotto: a) elaborazione congiunta
  - Fornitore: prevalentemente a) mandatario
  - Ripartizione dei costi: b) chiave di ripartizione

  Nota: l'elenco ufficiale dei Comuni GERES non è open source, tuttavia molti aspetti dell'organizzazione possono essere rilevanti anche per le applicazioni open source.
  Documenti: statuti (pubblici) e regolamento (su richiesta).

- iGov Portal

La presente sezione contiene al momento piani del Cantone di Berna e sarà integrata non appena saranno disponibili esempi della Confederazione.

E. Esempi di collaborazioni

In questa sezione sono esposti alcuni esempi che mostrano come sono state avviate delle collaborazioni.

E.1: Collaborazione: ZenDiS (Germania)

ZenDiS[16] è un'organizzazione che mira a ridurre la dipendenza dell'amministrazione pubblica (in particolare di quella tedesca) attraverso la promozione di OSS.

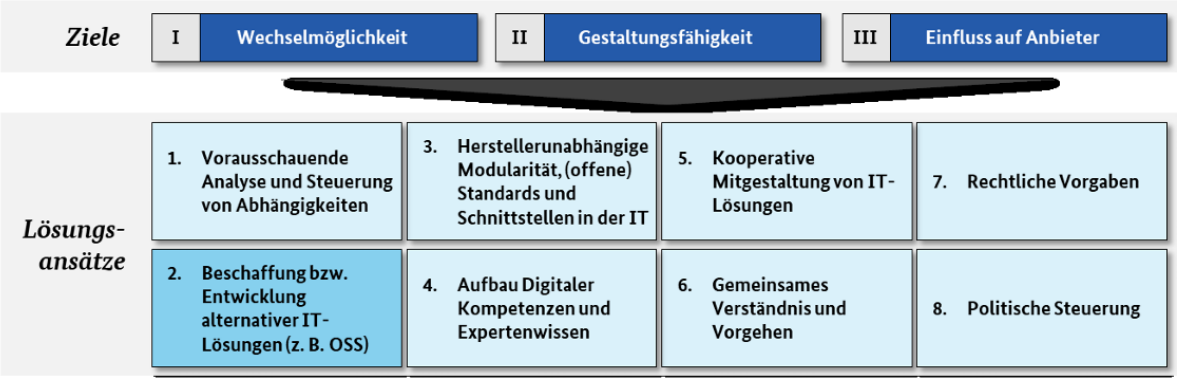

Figura 2: obiettivi e soluzioni per il rafforzamento della sovranità digitale (ZenDiS).

ZenDiS lavora a tal fine con partner strategici.

Nel caso dell'Amministrazione federale, la collaborazione non dovrebbe essere particolarmente stretta e il coordinamento non dovrebbe risultare vincolante; in ogni caso, questo potrebbe sempre cambiare. Concretamente, l'Amministrazione federale dovrebbe creare un'organizzazione propria per la Svizzera e coordinarsi in modo informale con ZenDiS. Questa operazione potrebbe avvenire tramite eOperations una volta presentata la relativa richiesta e stanziato un budget. Un'argomentazione a favore di questa organizzazione parallela è disponibile su eGovernment o nella LMeCA.
Attenzione: ZenDiS sottostà al diritto in materia di appalti pubblici dell'UE / della Germania. A causa delle considerevoli differenze con il diritto svizzero, ZenDiS non può essere riprodotta tale e quale in Svizzera e la collaborazione fra le due parti non è delle più semplici.

E.2: Collaborazione: Open Trip Planner

Open Trip Planner[17] (OTP) è un OSS per la pianificazione dei viaggi, gestito dalla Software Freedom Conservancy di New York. Gli sviluppatori sono finanziati da aziende di trasporto e fornitori. La particolarità è che ENTUR[18], attore statale di proprietà del Ministero dei trasporti e delle comunicazioni della Norvegia, ha assunto un ruolo di rilievo nell'ambito di OTP. ENTUR sviluppa principalmente per sé, ma ha è anche interessata a collaborazioni, in particolare per quanto riguarda i paesi nordici. In qualità di attore principale, si fa carico di una parte significativa dei costi di coordinamento. Chiunque può proporre nuove funzionalità, le quali vengono discusse in riunioni di coordinamento e sviluppate se qualcuno è disposto a farsi carico dei costi. Le singole funzionalità sono quindi finanziate principalmente dagli enti interessati; in altre parole, ognuno paga il servizio che richiede.
In Svizzera, anche i servizi di supporto dovrebbero essere nuovamente messi a concorso.

16 www.zendis.de/

17 www.opentripplanner.org/

18 https://entur.no/

E.3: Nuova collaborazione della Confederazione: trustbroker.swiss

Trustbroker.swiss è il software open source utilizzato per eIAM, l'infrastruttura di login centrale della Confederazione. Diversi Cantoni hanno deciso di utilizzarlo direttamente come SaaS o con proprie istanze. Dato che è disponibile come AGPL, la Confederazione può trarre vantaggio da eventuali estensioni. Tuttavia, per motivi di sicurezza è esclusivamente la Confederazione a occuparsi dello sviluppo del software nell'ambito dell'Amministrazione federale. Altre organizzazioni hanno la possibilità di richiedere nuove funzionalità, che vengono implementate se il team di progetto le considera utili.

E.4: Nuova collaborazione dei Cantoni: inosca

Alcuni Cantoni hanno collaborato per creare inosca[19], una comunità di sviluppo che si occupa delle autorizzazioni elettroniche. Il punto di partenza è stato eBau[20], che include anche Caluma[21], un framework per l'editing collaborativo. Inosca è un esempio di una parte di una soluzione che è stata trasferita in un proprio framework applicabile in modo più ampio. Seppure parzialmente soggetto a condizioni eccezionali, questo framework può essere utilizzato anche per altri progetti. In questo modo, molti altri enti hanno la possibilità di usufruirne.

La community si riunisce almeno una volta al mese per discutere di temi specializzati e pianificare insieme la roadmap. Tutti i partecipanti possono proporre dei temi. Se uno di questi è considerato utile per la community, i Cantoni interessati possono annunciarsi e organizzare insieme un gruppo di lavoro. I risultati ottenuti vengono costantemente riferiti alla comunità.
Inosca si basa sul principio che le nuove funzionalità vanno finanziate dai chi le progetta. Ciò significa che i costi vengono suddivisi fra i Cantoni che fanno parte del gruppo di lavoro sulla base di una chiave di ripartizione definita a priori dalla community. Una volta pronta all'uso, la funzionalità è messa a disposizione anche di tutti gli altri Cantoni senza spese aggiuntive.

Si è anche deciso di pianificare un piccolo importo forfettario annuale per le spese amministrative, che viene assegnato alle parti che si occupano dell'organizzazione della community.

E.5: Contributi diretti degli sviluppatori: SNOWPACK di WSL

SNOWPACK[22] è un progetto open source sviluppato da WSL per la modellizzazione della formazione del manto nevoso. Trattandosi di un modello altamente specializzato, la community del progetto è stata integrata in quella già esistente per questo ambito.
Alcune problematiche vengono segnalate come possibili contributi. Questo significa che ognuno si fa carico dei propri costi e contribuisce con le proprie funzionalità. In questo contesto vengono descritti anche lo stile di codifica e il processo di rilascio.

19 https://inosca.ch/

20 https://github.com/inosca/ebau

21 https://github.com/projectcaluma

22 https://snow-models.gitlab-pages.wsl.ch/snowpack-web/

F. Spunti per gli statuti di associazioni

Gli statuti dovrebbero essere strutturati nel modo più semplice possibile. Seguono esempi che possono essere presi come spunto per i propri statuti.

- eCH: www.ech.ch/sites/default/files/page/STAT_d_DEF_2014-04-10_ech-Statuten.pdf
- Stop Piracy: www.stop-piracy.ch/wp-content/uploads/2022/01/Statuten_d_10_09_2021.pdf

Verranno integrati ulteriori esempi non appena saranno disponibili.

G. Esempi di supporto ed emolumenti

- Emolumenti per prestazioni di servizi statistici: www.fedlex.admin.ch/eli/cc/2003/326/de
- Emolumenti IFPDT: www.edoeb.admin.ch/edoeb/it/home/datenschutz/grundlagen/gebuehren.html
- Prestazioni gratuite IPI per corsi in ambito di proprietà intellettuale (non legata ai software, ma solo come esempio): www.ige.ch/it/servizi/ip-academy/informazioni-generali/prezzi

Verranno integrati ulteriori esempi non appena saranno disponibili.